VIII Jornadas de Ingeniería Telemática

JITEL 2009

Cartagena

15 al 17 de septiembre de 2009

Escuela Técnica Superior de Ingeniería de Telecomunicación
Campus Muralla del Mar Antiguo Cuartel de Antigones, s/n
30202 Cartagena.
http://www.jitel09.upct.es/
e-mail: jitel09@upct.es

Editores:

Juan Carlos Sánchez Aarnoutse
Juan Pedro Muñoz Gea
Josemaría Malgosa Sanahuja

AQUILINE GRUPO

REGIÓN DE MURCIA fundaciónséneca AGENCIA REGIONAL DE CIENCIA Y TECNOLOGÍA

Universidad Politécnica de Cartagena

GOBIERNO DE ESPAÑA
MINISTERIO DE CIENCIA E INNOVACIÓN

Asociación de Ingeniería Telemática @t

VIII Jornadas de Ingeniería Telemática

JITEL 2009

Libro de ponencias

Cartagena, del 15 al 17 de Septiembre de 2009

Editores:
Juan Carlos Sánchez Aarnoutse
Juan Pedro Muñoz Gea
Josemaría Malgosa Sanahuja

ISBN: 978-84-96997-27-1
Editores: J. C. Sánchez Aarnoutse, J. P. Muñoz Gea, J. Malgosa Sanahuja, Universidad Politécnica de Cartagena

Organizan

Asociación de Ingeniería Telemática

Grupo de Ingeniería Telemática (UPCT)

Patrocinan

REGIÓN DE MURCIA
fundaciónséneca
AGENCIA REGIONAL DE CIENCIA Y TECNOLOGÍA

AQUILINE GRUPO

Escuela Técnica Superior de Ingeniería de Telecomunicación

Universidad Politécnica de Cartagena

GOBIERNO DE ESPAÑA

MINISTERIO DE CIENCIA E INNOVACIÓN

Comité de Programa

Josemaría Malgosa Sanahuja (Presidente) (Universidad Politécnica de Cartagena)

Jaume Abella Fuentes (Universitat Ramon Llull)
Javier Aracil Rico (Universidad Autónoma de Madrid)
Arturo Azcorra Saloña (Presidente Asociación de Telemática)
Víctor M. Carneiro Díaz (Universidade da Coruña)
Carlos Delgado Kloos (Universidad Carlos III de Madrid)
Jesús E. Díaz Verdejo (Universidad de Granada)
Yannis Dimitriadis (Universidad de Valladolid)
Rafael M. Estepa Alonso (Universidad de Sevilla)
Santiago Felici Castell (Universitat de València)
Julián Fernández Navajas (Universidad de Zaragoza)
Sebastián García Galán (Universidad de Jaén)
Mercedes Garijo Ayestarán (Universidad Politécnica de Madrid)
Ana Gómez Oliva (Universidad Politécnica de Madrid)
Antonio Gómez Skarmeta (Universidad de Murcia)
Jesús M. González-Barahona (Universidad Rey Juan Carlos)
José Luis González Sánchez (Universidad de Extremadura)
Victor Guillermo García García (Universidad de Oviedo)
Klaus Hackbart (Universida de Cantabria)
Xavier Hesselbach Serra (Universitat Politècnica de Catalunya)
Eduardo Jacob Taquet (Euskal Herriko Unibertsitatea)
Miguel Ángel López Carmona (Universidad de Alcalá)
Jorge Martínez Bauset (Universitat Politècnica de València)
Pedro Merino Gómez (Universidad de Málaga)
Daniel Morató Osés (Universidad Pública de Navarra)
Miquel Oliver Riera (Universitat Pompeu Fabra)
Magdalena Payeras Capellà (Universitat de les Illes Balears)
Manuel Ramos Cabrer (Universidade de Vigo)
Álvaro Suárez Sarmiento (Universidad de Las Palmas de Gran Canaria)

Comité Organizador

Josemaría Malgosa Sanahuja (Presidente)

Juan José Alcaraz Espín
María Victoria Bueno Delgado
Francesc Burrull i Mestres
María Dolores Cano Baños
Esteban Egea López
Juan García Haro
Antonio Javier García Sánchez
Felipe García Sánchez
Pawel Kulakowski
Fernando Losilla López
Pilar Manzanares López
Alejandro Santos Martínez Sala
Juan Pedro Muñoz Gea
Pablo Pavón Mariño
Juan Carlos Sánchez Aarnoutse

Revisores

Ramón Agüero (Universidad de Cantabria)
Mónica Aguilar Igartua (Universitat Politècnica de Catalunya)
Bernardo Alarcos (Universidad de Alcalá)
Juan J. Alcaraz Espín (Universidad Politécnica de Cartagena)
Florina Almenárez (Universidad Carlos III de Madrid)
Manuel Álvarez Díaz (Universidade da Coruña)
Luis Álvarez Sabucedo (Universidade de Vigo)
Pablo Ameigeiras (Universidad de Granada)
Mercedes Amor (Universidad de Málaga)
Ramón Aparicio Pardo (Universidad Politécnica de Cartagena)
Armando Astarloa Cuellar (Euskal Herriko Unibertsitatea)
Jasone Astorga Euskal (Herriko Unibertsitatea)
Alberto Banchs Roca (Universidad Carlos III de Madrid)
Jaume Barceló (Universitat Pompeu Fabra)
Boris Bellalta (Universitat Pompeu Fabra)
Fernando Bellas Permuy (Universidade da Coruña)
Carlos Jesús Bernardos (Universidad Carlos III de Madrid)
Miren Nekane Bilbao Marón (Euskal Herriko Unibertsitatea)
María Victoria Bueno Delgado (Universidad Politécnica de Cartagena)
Francesc Burrull i Mestres (Universidad Politécnica de Cartagena)
Fidel Cacheda Seijo (Universidade da Coruña)
Manuel Caeiro Rodríguez (Universidade de Vigo)
Celeste Campo (Universidad Carlos III de Madrid)
Cristina Cano (Universitat Pompeu Fabra)
María Dolores Cano Baños (Universidad Politécnica de Cartagena)
Víctor M. Carneiro Díaz (Universidade da Coruña)
Vicente Casares Giner (Universitat Politècnica de València)
Alberto Cortés (Universidad Carlos III de Madrid)
Rubén Cuevas Rumín (Universidad Carlos III de Madrid)
Enrique de la Hoz (Universidad de Alcalá)
Antonio de la Oliva (Universidad Carlos III de Madrid)
Javier de Pedro Cariacedo (Universidad de Alcalá)
Daniel Díaz (Universidad Carlos III de Madrid)
Jesús E. Díaz Verdejo (Universidad de Granada)
Almudena Díaz Zayas (Universidad de Málaga)
Mª José Domenech Benlloch (Universitat Politècnica de València)
Gerard Draper Gil (Universitat de les Illes Balears)
Esteban Egea López (Universidad Politécnica de Cartagena)
Antonio Estepa (Universidad de Sevilla)
Diego Fernández (Universidade da Coruña)
Gregorio Fernández (Universidad Politécnica de Madrid)
Julián Fernández Navajas (Universidad de Zaragoza)
Marcel Fernández Muñoz (Universitat Politècnica de Catalunya)
Pep Lluís Ferrer Gomila (Universitat de les Illes Balears)
Vreixo Formoso (Universidade da Coruña)
Jon Matías Fraile (Euskal Herriko Unibertsitatea)

Marta García (Universidad de Cantabria)
Nuria García (Universitat Pompeu Fabra)
Alberto García (Universidad de Cantabria)
Ana Belén García Hernando (Universidad Politécnica de Madrid)
Felipe García Sánchez (Universidad Politécnica de Cartagena)
Antonio Javier García Sánchez (Universidad Politécnica de Cartagena)
Pedro García Teodoro (Universidad de Granada)
Alberto García Martínez (Universidad Carlos III de Madrid)
Carlos García Rubio (Universidad Carlos III de Madrid)
Mercedes Garijo Ayestarán (Universidad Politécnica de Madrid)
José Manuel Giménez Guzmán (Universidad de Alcalá)
Ana Gómez Oliva (Universidad Politécnica de Madrid)
Antonio Gómez Skarmeta (Universidad de Murcia)
José C. González Cristóbal (Universidad Politécnica de Madrid)
Lluis Gutiérrez (Universitat Politècnica de Catalunya)
Ángela Hernández Solana (Universidad de Zaragoza)
Xavier Hesselbach Serra (Universitat Politècnica de Catalunya)
Mariví Higuero Aperribai (Euskal Herriko Unibertsitatea)
Xisca Hinarejos Campos (Universitat de les Illes Balears)
Gabriel Huecas (Universidad Politécnica de Madrid)
Llorenç Huguet Rotger (Universitat de les Illes Balears)
Jorge Infante (Universitat Pompeu Fabra)
José Irastorza (Universidad de Cantabria)
Mikel Izal (Universidad Pública de Navarra)
Eduardo Jacob (Euskal Herriko Unibertsitatea)
Sara Lana Serrano (Universidad Politécnica de Madrid)
David Larrabeiti (Universidad Carlos III de Madrid)
Fidel Liberal Malaina (Euskal Herriko Unibertsitatea)
Lourdes López (Universidad Politécnica de Madrid)
Javier López (Universidad de Málaga)
Cristina López Bravo (Universidade de Vigo)
Juan M. López Soler (Universidad de Granada)
Gabriel Maciá Fernández (Universidad de Granada)
Elsa Macías López (Universidad de Las Palmas de Gran Canaria)
Germán Madinabeitia (Universidad de Sevilla)
Eduardo Magaña (Universidad Pública de Navarra)
José Manuel Arco (Universidad de Alcalá de Henares)
Pilar Manzanares López (Universidad Politécnica de Cartagena)
José A. Mañas (Universidad Politécnica de Madrid)
Andrés Marín (Universidad Carlos III de Madrid)
Domingo Marrero Marrero (Universidad de Las Palmas de Gran Canaria)
Jesús Martínez Martínez (Universidad de Málaga)
Antonio Martínez Mas (Universidad Politécnica de Madrid)
Isaías Martínez Yelmo (Universidad Carlos III de Madrid)
Jorge Mata Díaz (Universitat Politècnica de Catalunya)
Vanesa Merchan (Universidad de Cantabria)
Pedro Merino Gómez (Universidad de Málaga)
Daniel Morató Osés (Universidad Pública de Navarra)
Mario Muñoz Organero (Universidad Carlos III de Madrid)
Juan Pedro Muñoz Gea (Universidad Politécnica de Cartagena)

Macià Mut Puigserver (Universitat de les Illes Balears)
José Luis Narbona Moreno (Universidad de Alcalá de Henares)
Andrés Navarro (Universidad de Alcalá de Henares)
Jorge Navarro Ortiz (Universidad de Granada)
Francisco Javier Novoa de Manuel (Universidade da Coruña)
Miquel Oliver (Universitat Pompeu Fabra)
Alberto Pan Bermúdez (Universidade da Coruña)
Abelardo Pardo Sánchez (Universidad Carlos III de Madrid)
Pablo Pavón Mariño (Universidad Politécnica de Cartagena)
Magadalena Payeras Capellà (Universitat de les Illes Balears)
Joseph Pegueroles Valles (Universitat Politècnica de Catalunya)
Emilia Pérez Belleboni (Universidad Politécnica de Madrid)
Francisco Pérez Vico (Universidad de Málaga)
Pedro José Piñero Escuer (Universidad Politécnica de Cartagena)
Vicent Pla (Universitat Politècnica de València)
Miguel Ángel Quintana Suárez (Universidad de Las Palmas de Gran Canaria)
Jaume Ramis Bibiloni (Universitat de les Illes Balears)
Manuel Ramos (Universidade de Vigo)
Juan J. Ramos Muñoz (Universidad de Granada)
Juan Raposo Santiago (Universidade da Coruña)
Felip Riera Palou (Universitat de les Illes Balears)
David Rincón (Universitat Politècnica de Catalunya)
Isabel Román (Universidad de Sevilla)
Gregorio Rubio Cifuentes (Universidad Politécnica de Madrid)
José Ruiz Mas (Universidad de Zaragoza)
Alberto Salmerón Moreno (Universidad de Málaga)
Luis Sánchez (Universidad de Cantabria)
Luis Sánchez Fernández (Universidad Carlos III de Madrid)
Juan Carlos Sánchez Aarnoutse (Universidad Politécnica de Cartagena)
Roberto Sanz (Universidad de Cantabria)
Isaac Seoane (Universidad Carlos III de Madrid)
Pablo Serrano Yañez-Mingot (Universidad Carlos III de Madrid)
Anna Sfairopoulou (Universitat Pompeu Fabra)
Marta Solera (Universidad de Málaga)
Álvaro Suárez Sarmiento (Universidad de Las Palmas de Gran Canaria)
Juan A. Ternero (Universidad de Sevilla)
Juan José Unzilla Galán (Euskal Herriko Unibertsitatea)
Miguel Ángel Valero Duboy (Universidad Politécnica de Madrid)
Juan Ramón Velasco (Universidad de Alcalá de Henares)
Iván Vidal (Universidad Carlos III de Madrid)
Víctor Villagrá (Universidad Politécnica de Madrid)
Juan Vozmediano (Universidad de Sevilla)
Johan Zuidweg (Universitat Pompeu Fabra)

Contenido

Sesión 1A: Redes Inalámbricas I.

Martes 15, 11:30 a 13:30

Sesión 1B: Seguridad en Internet.

Martes 15, 11:30 a 13:30

Sesión 2A: Redes Inalámbricas II.

Martes 15, 15:00 a 17:00

Sesión 2B: Servicios Telemáticos I.

Martes 15, 15:00 a 17:00

Sesión 3A: Redes Inalámbricas III.

Miércoles 16, 9:00 a 11:00

Sesión 3B: Servicios Telemáticos II.

Miércoles 16, 9:00 a 11:00

Sesión 4A: Comunicaciones Móviles.

Jueves 17, 9:00 a 11:00

Sesión 4B: Comunicaciones Multimedia.

Jueves 17, 9:00 a 11:00

Sesión 5A: Sistemas y Servicios Distribuidos.

Jueves 17, 11:30 a 15:30

Sesión 5B: Planificación y Gestión de Redes.

Jueves 17, 11:30 a 15:30

Sesión de Posters

Miércoles 16, 12:30 a 13:30

Influencia de la variabilidad del canal en un sistema de localización para interiores

Alejandro Martínez Sala, Raúl Guzman Quirós, Esteban Egea López
Departamento de Tecnologías de la Información y las Comunicaciones
Universidad Politécnica de Cartagena (UPCT)
Antiguo Cuartel de Antigüones (Campus Muralla del Mar), 30202 Cartagena
AlejandroS.Martinez@upct.es, Raul.Guzman.Quiros@gmail.com, Esteban.Egea@upct.es

***Resumen*-. Los sistemas de localización en interiores son la base de prometedores servicios telemáticos conscientes del contexto y entorno y se requiere un control de la posición con un error acotado. La medida de potencia de la señal recibida se puede usar como métrica para determinar la posición a partir de un algoritmo de localización. Se requiere hacer medidas para caracterizar los patrones de la señal recibida en un conjunto de puntos de interés (técnica de *fingerprint*). El canal en un interior es difícil de modelar y presenta un comportamiento aleatorio. En este trabajo de investigación se usan redes neuronales como herramienta matemática que aprende los patrones del *fingerprint* y su posición (x,y) asociada y son capaces de estimar una posición. Se evalúan varias arquitecturas del perceptron multicapa y se analiza el error de la estimación de la posición y su influencia con la variabilidad del canal para un escenario de una red WiFi.**

***Palabras Clave*- localización, entorno interior, canal estocástico, WLAN, red neuronal.**

I. INTRODUCCIÓN

Los sistemas de localización son claves para el desarrollo e implantación de servicios telemáticos contextuales y conscientes del entorno [1]. De especial relevancia son los servicios que se pueden ofrecer en entornos de interior; sistemas de ayuda y navegación para discapacitados, control y seguimiento de pacientes o equipos en hospitales, servicios contextuales en museos, aeropuertos, galerías comerciales, etc. Estos sistemas requieren de un sistema de posicionamiento que permita saber con un reducido y acotado margen de error su ubicación. La tecnología GPS no es adecuada para entornos de interior debido a la escasa o nula cobertura de los satélites de la red de posicionamiento por lo que hay que recurrir a otras tecnologías inalámbricas para la localización.

Cada vez es más frecuente que en un entorno objetivo haya desplegada una infraestructura de puntos de acceso (*Access Point*, AP) WiFi interconectados a una red Ethernet cableada. A su vez, los terminales móviles (*Mobile Stations*, MS) que portan los usuarios se van abaratando, aumentado su capacidad de proceso y una gran mayoría están dotados de una interfaz WiFi. Por tanto, el amplio despliegue y adopción de las redes WiFi supone que esta tecnología de comunicaciones sea un soporte idóneo para el desarrollo de servicios de localización en interiores.

Un sistema de posicionamiento calcula la posición a partir de parámetros de la señal generada entre un MS y varios AP, después procesa la señal medida mediante un algoritmo de localización que estima la posición del MS. Existen varias medidas que se pueden hacer de una señal de radiofrecuencia para determinar la posición de su emisor: *Angle of Arrival* (AoA), *Time of Arrival* (ToA) y *Received Signal Strength* (RSS) [1]. La medida del AoA y el ToA requiere de hardware especial y no es sencilla de conseguir con todo el hardware WiFi comercial. Sin embargo, lo más común, barato y disponible en cualquier equipo WiFi es la medición de la potencia de señal recibida (RSS).

En un entorno interior el canal inalámbrico a 2.4GHz, usado en el estándar IEEE 802.11, se comporta de forma estocástica y es complejo de caracterizar debido a la disposición física de los muros y obstáculos, el predominio de la propagación NLoS (*No Line of Sight*), es decir sin tener visión directa entre el emisor y el receptor, y efectos de desvanecimientos de la señal (*fading*) debidos al multicamino (*multipath*) y la propia naturaleza dinámica del entorno (movilidad de objetos y las propias personas). Por tanto se constata que la potencia de la señal recibida RSS no sigue un modelo determinista ideal sino que es una variable aleatoria difícil de modelar y dependiente del entorno y escenario de interés.

A su vez los algoritmos de localización se pueden clasificar como basados en la distancia o como un problema de reconocimiento de patrones. Así un ejemplo de algoritmo basado en distancia sería el basado en triangulación donde, a partir de un modelo de canal, se calcula la distancia relativa de un MS con al menos tres AP para estimar su posición. Debido a la naturaleza estocástica de un canal inalámbrico estos métodos producen errores bastante elevados y no son muy aconsejables para interiores.

Un algoritmo de localización fundamentado en reconocimiento de patrones usa una señal característica, como la RSS, que forma un patrón determinado en varios puntos específicos y conocidos. La técnica de medir los patrones característicos de la RSS se denomina RF *fingerprint* y requiere una campaña de medidas (proceso *offline* también denominado de calibración) para crear el "mapa radio" (*radio map*) del entorno de interés. Se constata que los puntos del *radio map* deben ser significativos y con una cantidad mínima para lograr una adecuada precisión. Durante la fase *online* de localización, la posición del MS se estima comparando la señal medida con el *radio map*. Existen varias propuestas de algoritmos basados en reconocimiento de patrones que usan la técnica del *fingerprint*: RADAR [2] es una de las primeras propuestas de sistemas de localización usando WiFi que usa el algoritmo

KNN (*K-Nearest Neighbors*) donde la señal muestreada de un transmisor se compara con todos los patrones del *radio map* buscando el patrón de referencia con la menor distancia Euclídea. EKAHAU [3] es el principal sistema de localización comercial WiFi que usa clasificadores bayesianos y técnicas estadísticas para inferir la posición. Battiti *et. al.* [4] propone el uso de una red neuronal del tipo perceptrón multicapa en un entorno de 640 m^2 con 3 AP usando medidas reales obtenidas de terminales WiFi; se fija y expone una única arquitectura de la red neuronal obteniéndose un error medio de 1.82 metros.

En el presente artículo se parte del trabajo de Battiti *et. al.* [4] pero estudiando y comparando el desempeño de varias arquitecturas del perceptrón multicapa y parámetros de aprendizaje en un entorno tipo oficina de mayor superficie (2500 m^2) con tres AP pero además del error medio, se usa la probabilidad del error que es una métrica más representativa y exigente para evaluar un sistema de localización. Finalmente, se analiza mediante simulación la influencia del canal en el aprendizaje de la red neuronal y en el error cometido, usando modelos del canal determinista hasta un canal aleatorio con una alta variabilidad.

El resto del artículo se estructura en los siguientes apartados. En la sección II se describe el modelo del entorno y del canal usado, así como el escenario de simulación. A continuación se definen las métricas para evaluar el desempeño del sistema. En la sección IV se explica y justifica la arquitectura del perceptron multicapa escogida y sus principales parámetros de configuración. Después en la sección V se presentan y analizan los resultados para finalizar con la sección VI de Conclusiones.

II. MODELADO DEL ENTORNO Y CANAL

Un objetivo del presente trabajo de investigación ha sido el desarrollo de una herramienta de simulación para estudiar y probar sistemas de localización basados en herramientas matemáticas de reconocimiento de patrones RSS usando la técnica de *fingerprint*. En este enfoque se ha querido evitar el tener que hacer tediosas campañas de medidas de la potencia de la señal recibida. Se pretende tener modelos fiables que marquen una tendencia de toda la capa física y, de una forma controlada, analizar la tendencia de la bondad y desempeño de la herramienta matemática de localización ante un canal estocástico y con un coste computacional asumible.

A. Modelado del canal WiFi en un entorno interior

Teniendo en cuenta los requisitos preestablecidos, se ha escogido un modelo de canal para el estándar IEEE 802.11 a 2.4GHz que está recomendado para la estimación de la potencia de la señal recibida [5,6]. Este modelo está basado en un modelo de *path-loss* a dos pendientes y se ha modificado añadiendo las pérdidas por atenuaciones debidas a obstáculos (como muros, puertas, ventanas, etc.) [7].

El modelo de *path-loss* consiste en una pérdidas por espacio libre, L_{FS} con pendiente de pérdidas α_1 para distancias menores que una distancia d_{BP} (denominado punto de ruptura o *Breaking Point*) y α_2 para distancias mayores del punto de ruptura. El modelo de *path-loss* de espacio libre se define como:

$$L_{FS}(d) = L_0 + 10\alpha_1 \log_{10}(d) + X \quad (1)$$

Para el caso general de *path-loss* tenemos la siguiente función de pérdidas con la distancia:

$$L(d) = \begin{cases} L_{FS}(d) + \sum L_{obs} + X, d \le d_{BP} \\ L_{FS}(d_{BP}) + 10\alpha_2 \log_{10}\left(\frac{d}{d_{BP}}\right) + \sum L_{obs} + X, d > d_{BP} \end{cases} \quad (2)$$

Donde L es la atenuación (en dB) a una distancia d entre un emisor y el receptor para el modelo de canal propuesto que tiene tres términos: a) componente de *path-loss* en dB, d es la distancia considerada como la línea que recorre el rayo directo (en metros), α_1, α_2 es la pendiente de pérdidas antes y después del denominado punto de ruptura o d_{BP} (en metros). b) Además se ha añadido un término de pérdidas por obstáculos que suma las pérdidas debidas a que la señal atraviese un muro, una puerta, etc. Dependiendo del tipo de obstáculo y material en [7] se definen los valores aconsejables de L_{obs} (dB). c) Finalmente la componente aleatoria *X* modela el efecto del *shadow fading* mediante una variable aleatoria gaussiana de media cero y varianza σ_x:

$$f_x(x) = \frac{1}{\sqrt{2\pi}\sigma_x} e^{-\frac{x^2}{2\sigma_x^2}} \quad (3)$$

B. Descripción de la herramienta de simulación

El simulador está implementado en Matlab y usa la toolbox de Redes Neuronales [8] para desarrollo del motor de localización.

La herramienta de simulación permite caracterizar y representar un entorno de interior formado por plantas, habitaciones, pasillos y obstáculos. Una vez representado se crea un sistema de referencia cartesiano del escenario donde se posicionan los AP.

A continuación se definen los parámetros del modelo de canal según la eq. (2). En este modelo hay una componente determinista, dependiente de la geometría y materiales de los muros, puertas y ventanas, y una componente aleatoria que modela la variabilidad del canal.

La capa física implementada en el simulador incorpora el modelo de *path-loss* con *shadowing*, las pérdidas de atenuación por obstáculos, la ganancia y diagrama de radiación de las antenas de los AP y MS y parámetros básicos del hardware como la potencia de transmisión y el umbral de sensibilidad.

Una vez que se definen los parámetros de la capa física se se seleccionan una serie de puntos en el escenario, ya sea por medio de un mallado o *grid*, generados de forma aleatoria o simplemente seleccionando manualmente un número de puntos secuenciales (que podrían representar la ruta seguida por un MS).

Seguidamente se pasa a una segunda fase que engloba dos cálculos; por cada punto de acceso y para cada punto generado sobre el layout 2D se realiza un cálculo de la componente determinista del modelo en el que se evalúan las pérdidas producidas por la distancia (según modelo de *path-loss* fijado) y por obstáculos atravesados (muros, puertas y otros). A su vez se realiza un segundo cálculo que incorpora posibles efectos del canal que añaden una componente aleatoria a la señal, como puede ser el efecto de *shadowing*.

El hecho de trabajar de esta manera permite realizar las simulaciones de forma más rápida al poder obtener sólo una vez la parte determinista, y sobre dicho cálculo aplicar de forma rápida varios modelos controlados con parámetros que

incorporen cualquier aleatoriedad a la señal. De esta forma se tiene una herramienta de simulación abierta para incorporar otros efectos y fenómenos (como la variabilidad del hardware, etc.).

C. Descripción del escenario

El escenario objeto de estudio emula un entorno de oficinas típicas de 50mx50m que incorpora tres habitaciones y un pasillo (ver figura 2). Se considera que las habitaciones quedan separadas por paredes interiores de tipo ladrillo, que incorporan atenuaciones de 3dB cada una, según se especifica en [7]. A su vez, estas habitaciones se consideran oficinas típicas que, según los parámetros recomendados en [5,6] tienen un coeficiente α_1=2 y α_2=3.5 con un cambio de pendiente a 5 metros. Como parámetros recomendados, la variable aleatoria del *shadowing* puede oscilar entre 3 y 5 dB.

En el escenario simulado se han considerado varios canales, donde la componente determinista es idéntica, y donde se varía la componente aleatoria entre 1, 3 y 6 dB. Esta componente aleatoria máxima de 6 dB persigue incorporar y ser una primera aproximación a otros efectos aleatorios como las irregularidades en el diagrama de radiación de la antena y la variabilidad del hardware.

La potencia de transmisión se fija a 0 dBm y el umbral de sensibilidad de los receptores en -100 dBm. Las antenas de los AP y MS se consideran isotrópicas. Por último, los AP se sitúan a una altura de 2.2 metros y los MS se consideran que están a una altura media de 1.5m.

III. MÉTRICAS DEL SISTEMA DE LOCALIZACIÓN

Mediante la herramienta de simulación se obtiene como resultado final un conjunto de puntos (x,y) reales y los correspondientes valores (x',y') estimados. Se usa la distancia Euclídea como métrica para comparar la posición real y la estimada con la herramienta matemática. Del procesamiento estadístico se obtienen la función de densidad de probabilidad (fdp) del error de la posición.

A. Mean Absolute Error (MAE)

A partir de la fdp se calcula la esperanza del error que se denomina *Mean Absolute Error* (MAE) o error medio. Este estadístico es un punto de partida para evaluar un sistema de localización, pero es insuficiente para analizar su desempeño.

B. Precisión

La precisión se define como la probabilidad del error. Es decir, dada una posición (x',y')estimada, se garantiza con una cierta probabilidad que la posición real se encuentra a distancia menor o igual al error. Dicho de otra forma, si se obtiene una precisión de E metros con una probabilidad P cuando la red estime una posición, el punto (x,y) real se encuentra contenido con probabiliad P en el área delimitada por el el punto estimado y un disco de radio E.

IV. ARQUITECTURA DE LA RED NEURONAL

Los modelos de redes neuronales son técnicas de autoaprendizaje que pueden resultar efectivas para resolver problemas de localización actuando como aproximadores universales. Una propiedad muy importante es que no se requiere un conocimiento previo de la geometría del entorno (disposición de las habitaciones, muros y obstáculos), el modelo de propagación del canal ni la posición de los AP.

En el sistema que se presenta partimos de que las medidas de potencia de señal recibidas de un terminal móvil desde diferentes puntos de acceso (al menos tres) aportan la información necesaria para determinar la localización de su posición dentro de un área de trabajo. La función no lineal existente en un canal real entre potencia y distancia puede ser modelada de forma aproximada mediante redes neuronales a partir de la optimización de ciertos parámetros (pesos) de la misma, garantizándose un buen aprendizaje de la relación entre entradas (potencias de señal) y salidas (posición (x,y)), resultando estimaciones de las coordenadas con errores permisibles.

Por tanto, el objetivo es conseguir que nuestra red neuronal sea capaz de estimar dichas coordenadas (x,y) a partir de tres entradas: un vector de potencias de la señal que reciben tres puntos de acceso (RSS_1, RSS_2, RSS_3) durante un mismo intervalo de tiempo.

Para afrontar este problema se ha planteado un tipo de red neuronal muy común en problemas de clasificación y aproximación de funciones denominado redes neuronales multicapa (del inglés *Multilayer Perceptron*, MLP). Estas redes requieren de una fase de entrenamiento mediante un aprendizaje supervisado a partir del *radio map* creado del entorno en la que tenemos un conjunto de entradas (medidas RSS en puntos (x,y) conocidos) y salidas deseadas (posiciones (x,y) reales del terminal móvil), para optimizar los pesos de la red en base a la minimización de una función de error, consiguiendo así reducir en la medida de lo posible el error de generalización.

El error de generalización es un concepto muy importante que define el comportamiento de la red neuronal ya entrenada ante ejemplos de entradas nunca antes vistos por la misma (no usados como muestras de entrada en la fase de entrenamiento). Es decir, el concepto habla de la capacidad de generalización de la red en términos de error ante muestras genéricas que pudiesen aparecer. A cada época de entrenamiento, se le introduce en un orden aleatorio el conjunto de muestras de entrenamiento, de manera que la red adapte los pesos cada vez para minimizar la función de error. Si se sobreentrena en exceso la red neuronal con demasiadas épocas se corre el riesgo de que la red adapte sus pesos demasiado bien a las muestras del conjunto usado para el aprendizaje y sea muy precisa para los ejemplos de entrenamiento, pero pierda capacidad de generalización cometiendo errores altos en la posición estimada ante patrones de entrada no vistos nunca antes. En cambio, si no se usara el suficiente número de épocas en el entrenamiento se puede llegar a cometer errores altos por falta de optimización de los pesos.

La arquitectura del perceptron multicapa utilizada se organiza como sigue: las entradas se procesan secuencialmente a través de las distintas capas ocultas de la red, donde cada unidad ("neurona") calcula un producto escalar entre un vector de pesos y el vector de salidas dado por la capa previa. Una vez calculado dicho producto escalar, se aplica una función de transferencia (función de activación de la neurona) cuya salida forma parte del vector de entradas de la siguiente capa. Las funciones de transferencia evaluadas son la tangente hiperbólica y la sigmoidal para las neuronas de la capa oculta. La función de activación de las neuronas de

la capa de salida es la función identidad (lineal) para no limitar los valores de salida obtenidos por la red.

Por tanto, las arquitecturas evaluadas tienen tres neuronas en la capa de entrada (cada una conectada a un AP), un número variable de capas ocultas (1, 2 y 3) con un mismo número de neuronas en cada capa (desde 4 hasta 128 neuronas) y dos neuronas en la capa de salida que devuelven la coordenada X y la coordenada Y de la posición.

Para el entrenamiento de la red se ha utilizado un algoritmo iterativo basado en el método tradicional de Newton que se denomina algoritmo Levenberg Marquardt [9], el cual ha demostrado presentar un buen rendimiento para el entrenamiento de este tipo de redes en otros problemas de aproximación de funciones. Este algoritmo se basa en el uso del gradiente para minimizar la función de error usando la técnica de retropropagación (el inglés *back-propagation*) la cual se explica en el capítulo 11 de [10].

Respecto a la función de error a minimizar en el entrenamiento se ha usado el error cuadrático medio (MSE), definido como:

$$MSE = \sum_{n=1}^{N} (t_n - o_n(\omega))^2 \qquad (4)$$

Donde se observa como la función depende tanto de la diferencia entre las salidas estimadas por la red que es función de los pesos ω en ese instante, como de las salidas deseadas (también denominadas *target*) asociadas a la entrada.

El número de muestras de entrenamiento es un parámetro importante porque refleja cuantos datos se necesitan para que una red pueda aprender adecuadamente las características y particularidades de los patrones RSS del entorno y así aproximar la posición del terminal móvil.

V. METODOLOGÍA DE TRABAJO

En el trabajo de investigación realizado se emplea una herramienta de simulación y en la metodología de trabajo se han seguido las siguientes fases:

A. Caracterización del entorno y layout 2D

En este paso se define el *layout 2D* indicando las habitaciones y obstáculos existentes. Se genera un sistema de referencia cartesiano del entorno. Finalmente para dicho entorno se crea una capa de puntos de acceso que se fijan con unas coordenadas (x,y) determinadas y se les da un atributo de altura, necesario para calcular la distancia Euclídea con un MS.

B. Parámetros del modelo de capa física

En esta fase se ajustan los valores del modelo de *path-loss* para cada una de las zonas del entorno. En especial se configura la variabilidad del canal debida al *shadowing* fijando el valor de la componente gaussiana expresada en dB. A su vez, se configura el tipo de muro (según el material y el grosor) y se definen los parámetros de las pérdidas de atenuación (en dB) causadas cuando la señal atraviesa dicho muro. Finalmente, se fija el diagrama de radiación de las antenas de los AP y MS, la potencia de transmisión y el umbral de sensibilidad del hardware WiFi.

C. Cálculo de coberturas de los AP y generación de datos de aprendizaje y testeo

Para el cálculo de la cobertura que ofrecen los AP en el entorno una vez fijados todos los parámetros del modelo (tanto del canal, como del hardware y los parámetros de las antenas del terminal como de cada AP) se genera un mallado (*grid*) de puntos fijando una distancia mínima de separación (resolución del *grid*) entre los mismos. A continuación se calcula la potencia que recibe cada AP desde un supuesto terminal colocado en cada uno de dichos puntos del *grid* obteniéndose la cobertura que ofrece cada AP en el área total evaluada en los puntos del *grid*.

Para la generación de campañas de datos tanto para entrenar la red como para testearla posteriormente, se fija un número total de puntos que se generan de forma aleatoria intentando cubrir todo el mapa interior de forma que el conjunto de muestras final sea suficientemente representativo de todo el escenario bajo estudio. Posteriormente, y al igual que para el cálculo de la cobertura, se calcula la potencia recibida por cada AP según el modelo de canal parametrizado.

Las muestras de aprendizaje emulan una campaña de medidas para generar el *radio map* según la técnica del *fingerprint* explicada anteriormente. Se genera un *radio map* para el caso de un canal determinista (variable aleatoria del *shadowing* de 0 dB), y para los casos de un canal con una variabilidad de 1 dB, 3 dB y 6 dB. Por tanto, cada *radio map* generado refleja la tendencia de un canal con nula variabilidad (determinista) hasta una alta variablidad (*shadowing* de 6 dB).

A su vez, fijada la variable aleatoria gaussiana del canal, se genera un fichero de testeo con 800 muestras totalmente independientes y diferentes a las muestras de aprendizaje.

D. Comprobación de la red neuronal y extracción de los estadísticos de localización

Una vez que la arquitectura de la red neuronal converge a una solución, se le pasa el fichero de testeo y se compara cada punto la posición (x,y) real con la posición (x',y') estimada por la red. Se obtienen los estadísticos de error medio, la probabilidad de error al 95% y al 80%.

E. Análisis de resultados, tendencias y replanteamiento de hipótesis

Se analizan los resultados y se observan las tendencias en los estadísticos del error y la correlación con la arquitectura de la red y con los principales parámetros de configuración. Se vuelven a plantear hipótesis sobre el efecto de una arquitectura (número de capas ocultas y neuronas en capa oculta) y/o combinación de parámetros en la tendencia del error de generalización y se repiten los pasos previos.

F. Verificación y validación de los resultados

Una vez que una tendencia se ha contrastado para una determinada arquitectura y parámetros de configuración se vuelve a repetir el proceso de simulación pero variando hasta tres veces el *radio map* de entrenamiento y/o las muestras de testeo. Se debe volver a verificar la misma tendencia para contrastar y validar los resultados.

VI. RESULTADOS Y ANÁLISIS

Se han realizado 6400 redes entrenadas barriendo distintos parámetros de la arquitectura de la red como el número de capas ocultas, el número de neuronas por capa oculta o la función de transferencia de cada neurona combinado con distintos parámetros de la fase de entrenamiento de la red como el número de épocas o el número de muestras de entrenamiento de la red.

En aras de centrarse en los datos y resultados más relevantes, se resumen a continuación las principales pruebas y conclusiones obtenidas para descartar ciertas configuraciones y no explicarlas de forma extensa en los resultados.

Normalización de las señales de entrada a los AP

En la fase de entrenamiento se ha comprobado que la normalización de las entradas de los valores RSS en un rango de valores entre [-1,1] (a partir del valor máximo medido de -29dBm y mínimo en el umbral de sensibilidad de -100dBm) era positiva de cara a la minimización del error de generalización y la convergencia del mismo en el entrenamiento. Por tanto los resultados se focalizan en el caso de datos normalizados.

Arquitecturas de una capa con 4, 8, 16 y 32 neuronas

Se ha comprobado que las arquitecturas que menores errores de generalización han obtenido eran de sólo una capa oculta. Con arquitecturas de 2, 3 y 4 capas no se conseguían mejores resultados por lo que no se van a exponer estos casos. Además, se ha comprobado que opciones de 64, 128 ó 256 neuronas en capa oculta no aportan ventajas sustanciales e incrementan bastante el coste computacional.

Función de transferencia tangente hiperbólica

Se ha comprobado que el uso de una función de transferencia del tipo tangente hiperbólica $f(x) = (e^x - e^{-x})/(e^x + e^{-x})$, ofrece en general, mejores resultados para este problema en comparación con la función sigmoide, mientras que en la capa de salida se ha mantenido la función identidad para no limitar la salida.

Número de épocas

Para valores altos de épocas no se ha producido sobreentrenamiento. Se ha observado como para este problema el uso de entrenamientos con menos de 80 épocas y números de muestras entre 400 y 800 las redes conseguían errores de generalización aceptables y muy parecidos.

A. Resultados obtenidos

Los datos que se presentan a continuación están seleccionados para los siguientes rangos:

- Número de neuronas de la capa oculta: 4, 8, 16, 32.
- Variabilidad del canal: 0 dB, 1 dB, 3 dB, 6 dB.
- Número de muestras del *radio map*: 50, 100, 200, 400, 800.

En las tablas I-IV se muestra una selección de datos de las arquitecturas y configuraciones de interés. Se observa que para pocas muestras de aprendizaje, una red de 4 neuronas se comporta mejor que las redes con más neuronas aunque el error medio es significativo. Una tendencia que se empieza a observar es que conforme aumenta la aleatoriedad del canal, el error medio aumenta pero no de una forma significativa.

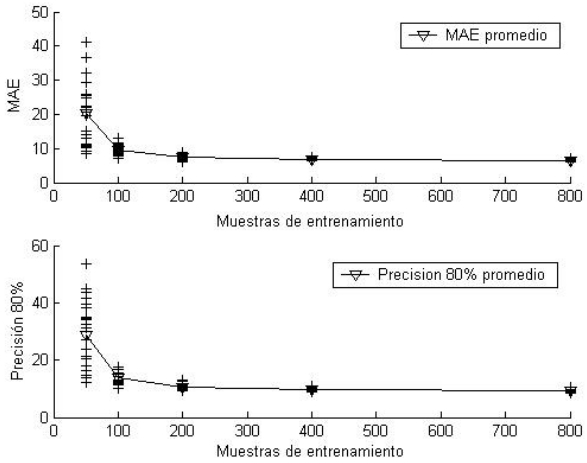


Fig. 1. Error medio y precisión vs. número de muestras de entrenamiento

En la figura 1 se observa la influencia del número de muestras en el error de generalización y en la probabilidad del error para un escenario de 16 neuronas y un canal con 6 dB de variabilidad. Es interesante destacar que para pocas muestras se obtiene un error medio mayor de 20 metros y una precisión de 30 metros con un 80% de probabilidad (es decir cuando la red devuelva un resultado con una probabilidad del 80% el punto estará contenido en un radio de 30 metros). Conforme aumenta el número de muestras el error medio se estabiliza entorno a 10 metros y una precisión a 9 metros con un 80% de probabilidad.

Esta tendencia se observa también en las tablas I-IV donde a partir de 200 muestras no se producen cambios significativos ni en los errores ni en la precisión obtenida para todas las redes neuronales y para los canales evaluados. Esta relación del número de muestras de entrenamiento óptimo tiene implicaciones en el tamaño del *radio map* y el tiempo que hay que invertir en hacer la campaña de medidas en un entorno real y requiere un estudio en mayor profundidad

Por tanto, a partir de 200 muestras se observa que la red aprende bastante bien y se estabiliza en un error medio entorno a 6 metros. Además, con una probabilidad del 80% cualquier estimación que haga la red se encuentra en un disco de radio 8-9 metros, y con una probabilidad del 95% ese disco tiene un radio de 15-18 metros.

Un efecto interesante es que cuando aumenta la variabilidad del canal (desde 0 a 6 dB) el error medio y la precisión son estables con variaciones reducidas.

En la siguiente figura se genera una trayectoria secuencial de puntos (círculos) por las habitaciones; en cada punto de la ruta el simulador calcula la RSS en cada AP para un canal con variabilidad de 6 dB. A continuación este vector de muestras RSS de los AP se pasa por una red de 32 neuronas entrenada con 800 muestras (error promedio de 6.9 metros, precisión al 80% y 95% de 9.9 y 16.7 metros respectivamente). Con aspas se representa la trayectoria estimada por la red y se puede observar que hay bastante coincidencia entre el valor real y el estimado en determinadas zonas del mapa.

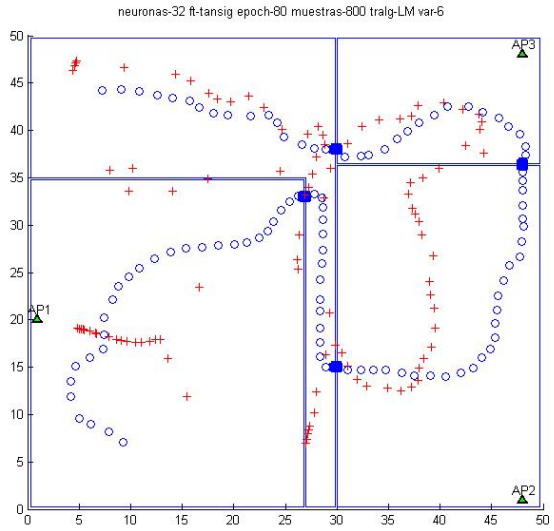


Fig. 2.Escenario con 3 habitaciones, un pasillo y 3 AP. Trayectoria real y estimada por una red entrenada con 32 neuronas y un canal con variabilidad de 6 dB.

VII. CONCLUSIONES

En el presente trabajo de investigación se ha desarrollado un simulador que permite caracterizar un entorno interior y la capa física que incluye el hardware, el diagrama de radiación de las antenas y un modelo de canal basado en *path-loss* de varias pendientes, pérdidas de atenuación por obstáculos y *shadowing*. El sistema propuesto permite definir tendencias complejas en el comportamiento del canal mediante modelos fáciles de implementar y computacionalmente abordables y recrear *radio maps*. El simulador es el soporte de para evaluar herramientas matemáticas de reconocimiento de patrones usando la técnica de RSS *fingerprint*.

Se han evaluado varias arquitecturas de redes neuronales basadas en el perceptron multicapa y se ha comprobado que se obtienen unos resultados y tendencias similares con una capa oculta y un número reducido de neuronas (entre 4 y 32). La función de activación en la capa oculta usada que ofrece mejores resultados es la tangente hiperbólica. A su vez, se ha comprobado que normalizar los valores de RSS de las muestras de entrenamiento hace que la red aprenda mejor.

Un resultado interesante es que conforme aumenta la variabilidad del canal (de 0 a 6 dB), la red filtra la componente aleatoria y los errores promedio y la precisión devuelven valores similares. Por tanto, un resultado preliminar a remarcar es que la red tiende a ser robusta al canal aleatorio, pero este punto hay que validarlo con un estudio en mayor profundidad modificando más parámetros de la capa física.

Para simulaciones de una variabildad del canal de 6 dB, 200 muestras y 32 neuronas se obtiene un error medio de 6.9 metros y una precisión del 80% de 9.9 metros, es decir, cuando la red entrenada devuelva una estimación con una probabilidad del 80%, el punto estimado estará contenido en un disco de radio 9.9 metros. Battiti *et al.* [4] en un escenario similar con 3 AP obtiene un error medio de 1.8 metros, pero no da datos sobre la precisión y hay que tener en cuenta que el entorno que estudia tiene 640 m^2 (cuatro veces más pequeño que el escenario de simulación).

Se ha comprobado que para el entorno emulado existe un número óptimo de 200 muestras a partir del cual la red neuronal converge a un error de generalización y no se obtienen mejoras significativas.

Como trabajos futuros se contempla analizar otros escenarios con distintas dimensiones y número de habitaciones, aumentar la complejidad del modelo de la capa física y completar el estudio del número de muestras óptimas para que la red obtenga un error de generalización acotado.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el proyecto nacional TEC2007-67966-01/TCM (CON-PARTE-1) y el proyecto regional SERPA (referencia 2107SE0027) y está también enmarcado en el "Programa de Ayudas a Grupos de Excelencia de la Región de Murcia, Fundación Séneca.

REFERENCIAS

[1] K. W. Kolodziej, J. Hjelm, *Local Positioning Systems: LBS Applications and Services, ed.* CRC, 2006
[2] P. Bahl, V.N. Padmanabhan, *RADAR: An in-building RF-based user location tracking system*, IEEE INFOCOM, pages 775-784, 2000
[3] EKAHAU, http://www.ekahau.com
[3] R. Battiti and A. Villani and T. Le Nhat, *Neural network models for intelligent networks: deriving the location from signal patterns,* in Proceedings of Annual Symposium on Autonomous Intelligent Networks and Systems, 2002
[5] V. Erceg et al., *TGn Channel Models*, IEEE 802.11 document 03/940r3, Mayo 2004
[6] J. Medbo, P. Schramm, *Channel models for HIPERLAN/2,* ETSI/BRAN document no. 3ERI085B
[7] T. Rappaport, *Wireless Communications: Principles and Practice,* ed. Prentice Hall, 2001
[8] MATLAB Neural Tool Box; http://www.mathworks.com/
[9] M.Hagan, M.Menhaj,"Training *Feedforward networks with the Marquardt algorithm*, IEEE T. Neural Networks 5 pag 989-993, 1994
[10] R.Want and B.Schilit, *Expanding the horizons of location-aware computing*, *IEEE Computer*, 34(8):31-34, August 2001

Tabla I ERROR Y PRECISIÓN PARA 50 MUESTRAS ENTRENAMIENTO Y 80 ÉPOCAS

	Componente aleatoria canal: 0dB				**Componente aleatoria canal: 1dB**				**Componente aleatoria canal: 3dB**				**Componente aleatoria canal: 6dB**			
N	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**
MAE	11.8	10.7	12.2	22.5	7.04	8.3	19.8	32.9	27.1	14.3	17.5	36.5	8.20	11.5	14.9	27.2
P80%	13.4	13.2	32.6	65.2	16.9	24.3	24.4	45.9	11.8	17.9	28.0	51.3	12.8	17.8	21.6	43.4
P95%	52.5	33.0	32.6	65.2	16.9	24.3	70.3	114.5	26.0	52.0	46.1	155.2	19.1	34.2	50.2	88.7

Tabla II ERROR Y PRECISIÓN PARA 100 MUESTRAS ENTRENAMIENTO Y 80 EPOCAS

	Componente aleatoria canal: 0dB				**Componente aleatoria canal: 1dB**				**Componente aleatoria canal: 3dB**				**Componente aleatoria canal: 6dB**			
N	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**
MAE	7.8	6.6	10.4	12.0	6.5	6.0	12.7	14.9	6.0	6.7	8.5	11.8	6.8	9.1	8.8	25.7
P80%	11.9	9.2	14.4	16.1	10.2	8.9	11.5	18.4	9.1	10.7	11.8	17.8	10.1	11.8	12.9	36.4
P95%	18.1	16.4	29.3	42.1	17.0	18.3	36.4	51.2	16.1	16.7	27.5	34.5	15.8	28.9	21.9	85.4

Tabla III ERROR Y PRECISIÓN PARA 200 MUESTRAS ENTRENAMIENTO Y 80 EPOCAS

	Componente aleatoria canal: 0dB				**Componente aleatoria canal: 1dB**				**Componente aleatoria canal: 3dB**				**Componente aleatoria canal: 6dB**			
N	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**
MAE	6.5	6.0	6.8	11.3	6.1	5.5	5.6	7.4	5.7	5.0	5.6	5.7	6.6	6.2	7.5	8.8
P80%	9.6	8.6	10.0	15.4	9.6	8.2	8.8	10.6	9.0	7.4	8.3	8.6	9.2	9.5	10.1	13.5
P95%	16.9	15.8	18.7	36.8	15.9	16.1	15.5	23.7	14.8	15.8	16.9	17.4	16.1	16.0	20.8	24.1

Tabla IV ERROR Y PRECISIÓN PARA 400 MUESTRAS ENTRENAMIENTO Y 80 EPOCAS

	Componente aleatoria canal: 0dB				**Componente aleatoria canal: 1dB**				**Componente aleatoria canal: 3dB**				**Componente aleatoria canal: 6dB**			
N	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**
MAE	5.9	5.7	6.2	6.3	6.1	5.5	5.6	6.6	5.8	5.1	5.0	5.1	6.6	6.0	7.3	6.9
P80%	9.4	8.1	9.7	9.2	8.9	8.2	8.4	10.3	8.8	7.9	7.7	8.0	9.3	8.4	10.6	9.9
P95%	15.7	16.9	15.7	18.7	15.2	15.7	15.6	18.7	14.9	15.5	15.4	17.5	16.6	17.3	18.0	16.7

CSMA/ECA: Carrier Sense Multiple Access with Enhanced Collision Avoidance

Jaume Barcelo
Universitat Pompeu Fabra
jaume.barcelo@upf.edu

Alberto Lopez Toledo
Telefonica Research
alopezt@tid.es

Cristina Cano
Universitat Pompeu Fabra
cristina.cano@upf.edu

Miquel Oliver
Universitat Pompeu Fabra
miquel.oliver@upf.edu

***Abstract*—This paper presents CSMA/ECA, which combines the efficiency of reservation-based protocols and the simplicity of random access mechanisms. The maximum efficiency of CSMA/CA with optimal parameter adjustment is easily exceeded by CSMA/ECA, even when fixed parameters are used by the latter. CSMA/ECA stations fairly coexist with legacy CSMA/CA and increase the portion of time that is devoted to successful transmissions while decreasing the number of collisions and empty slots. We show that the performance of CSMA/ECA is also clearly superior to the legacy one in lossy channels. The proposed mechanism initially behaves as a CSMA/CA network, but it progressively converges to a collision-free deterministic operation. The convergence process can be modelled as a Markov Chain to assess the duration of the transitory phase.**

I. Introduction

In many communications systems, a broadcast channel is shared by a set of stations. There are different strategies to arrange the sharing, which are called multiple access mechanisms. One option is to divide the resources (time, frequency, carriers or codes) among the different participating nodes. The nodes can also take turns in transmitting, and explicitly signal the end of each turn. Those alternatives prevent that two stations simultaneously transmit.

A popular medium access technique in local area networks is Carrier Sense Multiple Access (CSMA) [1]. The key property of CSMA networks is that the stations listen before transmitting. A station with data ready to transmit senses the channel for a given amount of time and, if the channel is detected idle, the station transmits. It is still possible that collisions occur in CSMA because the propagation of the communication signals is not instantaneous, and real communication systems require a certain amount of time to switch from a listening mode to a transmitting mode.

In CSMA with Collision Avoidance (CSMA/CA), the stations defer their transmission a random number of slots. The efforts to reduce the number of collisions are motivated by the fact that collisions represent a significant waste of resources in wireless networks, since it is not feasible to immediately detect a collision and interrupt the transmission. The stations either transmit or receive, and cannot collect any feedback from the radio channel while they are transmitting.

CSMA/CA combined with truncated Binary Exponential Backoff (BEB) is at the core of the Medium Access Control (MAC) specification in the suite of protocols IEEE 802.11 [2]. These protocols are widely used in Wireless Local Area Networks (WLANs) and, for this reason, they have been the subject of extensive research with the goal of reducing collisions and improving performance.

In spite of the possibility of collisions, CSMA/CA is still an appealing protocol for WLANs. It is lightweight, it takes advantage of statistical multiplexing to accommodate bursty traffic and it can be executed in a distributed fashion. CSMA/CA is especially fitted for networks with a large number of stations that sporadically send one packet. However, CSMA/CA was not designed to benefit from the fact that some stations have multiple-packet messages [3], [4], *i.e.* stations that store several packets in their transmission queues.

When stations send multiple consecutive packets, it is possible to use the feedback obtained from previous transmissions attempts to adequately schedule future transmissions. For this reason, we suggest a modification to the CSMA/CA protocol that further reduces the number of collisions while maintaining all its versatility and power. We call the new protocol CSMA with Enhanced Collision Avoidance (CSMA/ECA).

The main features of the presented CSMA/ECA protocol are the following:

- The maximum theoretical performance of CSMA/CA can be exceeded using CSMA/ECA.
- It provides a collision-free medium access after a transitory phase.
- It fairly coexists with legacy CSMA/CA.
- It works in a distributed fashion.
- It does not require additional computational efforts and can be easily implemented.
- It is robust against channel errors.

The rest of the paper is organized as follows: Section II defines the CSMA/ECA algorithm, then in Section III a Markov Chain model to predict the length of the transitory phase is described. Implementation issues and the performance evaluation results are discussed in Section IV while Section V presents an overview of the related work in the area. Finally, some conclusions are given.

II. Enhanced Collision Avoidance

In CSMA/CA, whenever there are backlogged stations with a packet ready to be transmitted, the channel time is implicitly divided into slots. Three different kinds of slots are differentiated: empty, successful and collision. A slot is empty when no station attempts transmission; successful if one (and only

one) station transmits; and collision if more than one station simultaneously transmit. The channel time spent in empty slots or collision slots is wasted.

Whenever a station has to defer its transmission, it chooses a random backoff value B from a contention window.

$$B \sim \mathcal{U}[0, CW - 1], \quad (1)$$

where $\mathcal{U}$ is the uniform distribution and CW is the contention window.

We consider that the stations are saturated (*i.e.* the stations always have a packet ready to transmit). As a consequence, the stations are either transmitting, receiving or backing off; they are never idle. After each transmission attempt, the stations choose a backoff value. The stations have to backoff both after collisions and successful transmissions. For the first case, the backoff has to be necessarily random to prevent a new collision in the retransmission attempt. However, for the second case, the backoff value can be deterministically selected.

A. Deterministic Backoff After Successful Transmissions

By choosing a deterministic backoff after a successful transmission and a random backoff otherwise, the system converges to a collision-free operation when the number of active stations is not greater than the value of the deterministic backoff. In the case of a successful transmission, the deterministic behaviour stabilizes the system (hopefully leading to another success). Conversely, if there is a collision, the randomness of the backoff provides a change that would (desirably) avoid more collisions. The system exploits the information gathered from previous transmission attempts to further reduce the collisions, thus we call it Enhanced Collision Avoidance (ECA). The terminals perform a random search to find free slots, until collisions disappear.

It has to be clear that a station keeps using a deterministic backoff after each successful transmission, until a collision occurs. As soon as it suffers a collision, the station moves back to the random behaviour. The collision will always be caused by a station that randomly selected its transmission slot, since collisions among stations that behave deterministically are not possible.

This principle can be better understood by means of an example. Consider the simplest case of two stations (*STA 0* and *STA 1*) contending for a channel, as shown in Fig. 1. The channel time advances from left to right and it is divided in slots. Even though the actual duration of empty, successful and collision slots differ, all the slots are equally represented in Fig. 1 for simplicity reasons. The upper channel time line corresponds to legacy CSMA/CA, while the lower one incorporates the modifications we have proposed for CSMA/ECA. The deterministic backoff after successes is a value that depends on the 802.11 flavor, as will be explained in subsection IV-A. In the example depicted in Fig. 1, a value equal to sixteen is used.

In the figure, the transmission attempts are represented as shaded boxes. Additionally, the figure also shows the backoff value chosen by each station (between brackets). The label indicates whether the backoff has been chosen randomly or deterministically.

In the example, the two CSMA/CA stations collide, then successfully transmit and, finally, collide again. When CSMA/ECA is used, collisions disappear after all stations have successfully transmitted, because the backoff is selected deterministically. It is useful to imagine a virtual frame[1] of V slots (represented with a dotted line in the figure) and observe that, after collisions disappear, the stations transmit in fixed slot positions within the virtual frame, similarly to a TDMA operation.

Algorithm 1 represents the protocol that is distributedly executed in each of the contending stations. The meaning of each of the variables is as folows:

- b is the backoff counter.
- CW_{min} is the minimum contention window.
- CW_{max} is the maximum contention window.
- a is the number of transmission attempts.
- A is the maximum number of transmission attempts.
- V is the deterministic backoff value after successful transmissions.

```
   /* Initialize b.                                  */
1  b ← U[0, CW_min − 1];
2  while there is a packet to transmit do
       /* Initialize a.                              */
3      a ← 0 ;
4      while a < A do
           /* First, backoff.                        */
5          while b > 0 do
6              wait 1 slot ;
7              b ← b − 1 ;
8          end
9          Attempt transmission ;
10         if success then
               /* Deterministic backoff.             */
11             b ← V;
12             break ;
13         else
               /* If transmission fails.             */
14             a ← a + 1 ;
               /* Random backoff value.              */
15             b ← U[0, min(CW_min * 2^a, CW_max) − 1];
16         end
17     end
18 end
```

Algorithm 1: CSMA/ECA

B. Efficiency of CSMA/ECA during Steady-State Operation

Let us define the channel efficiency (ϕ) as the fraction of channel time that is devoted to successful transmissions,

[1] Some works refer to data-link layer PDUs as frames. In this article, a frame is a group of slots. Data-link layer PDUs are called packets.

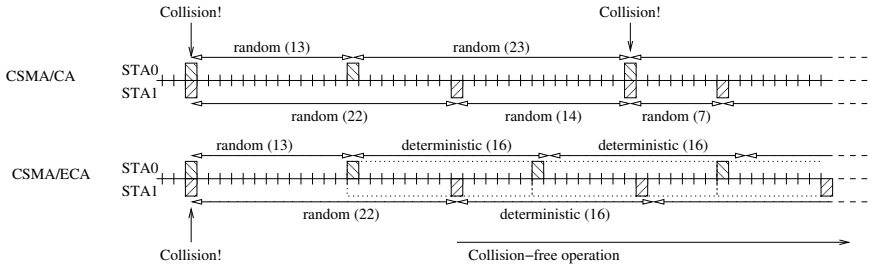


Fig. 1. CSMA/CA is compared to CSMA/ECA in an example in which two saturated stations contend for the channel. When CSMA/ECA is used, after both stations have successfully transmitted, the behaviour of the stations is deterministic and no more collisions occur.

$$\phi = \frac{P_s T_s}{P_e T_e + P_s T_s + P_c T_c}, \tag{2}$$

where P_e, P_s and P_c are the empty, success and collision probabilities, respectively. And T_e, T_s and T_c are the duration of an empty, successful and collision slot, respectively.

Then, for a number of contending stations (ς) not greater than the size of the virtual frame, the efficiency that can be obtained from CSMA/ECA in steady-state collision-free operation is :

$$\phi = \frac{\varsigma \cdot T_s}{\varsigma \cdot T_s + (V - \varsigma) \cdot T_e} \; ; \; \varsigma \leq V. \tag{3}$$

Fig. 2 compares the efficiency obtained with CSMA/ECA (solid line) with the efficiency delivered by CSMA/CA (dotted line). The upper theoretical maximum of CSMA/CA with perfect parameter adjustment is also included in the figure (using a dashed line). Finally, average simulation results of CSMA/ECA that include both transitory and stationary operation are represented as thick dots with bars to account for the 95% confidence interval. The IEEE 802.11b standard has been assumed, together with a data rate of 2 Mbps and a packet size equal to 1500 bytes. The efficiency is represented for an increasing number of contending stations. The concept of active (or contending) station deserves some further clarification. In a typical network, some of the stations are actively sending data while others are idle. Usually, the number of active stations is only a fraction of the stations that are registered to a given access point. As an example, if 50 stations with an activity rate of 10% share a given frequency band, the expected number of simultaneous contenders is 5.

The efficiency of CSMA/ECA is computed as presented in (3) using $V = 16$, the upper bound for CSMA/CA with dynamic parameter adjustment is obtained from [5], and the performance of plain CSMA/CA is computed using the approach in [6]. The simulation results are obtained using a custom simulator[2] in Octave, that includes only the MAC layer.

Before reaching the steady-state and obtaining the efficiency as presented in (3), the system goes through a transitory operation. The efficiency obtained in the transitory operation is a value between the efficiency delivered by CSMA/CA and

[2]All the source code is available upon request to the first author.

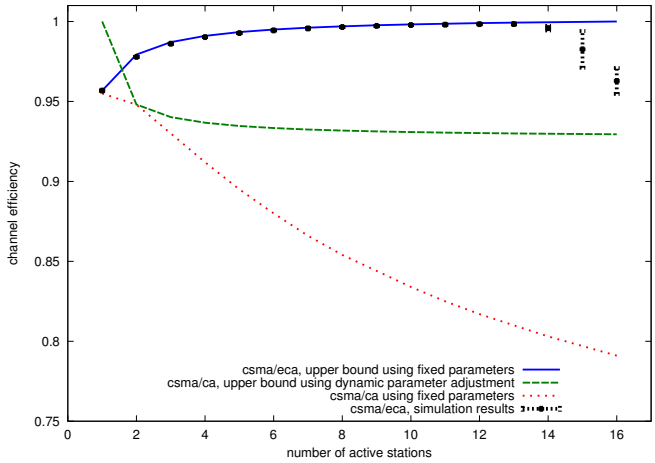


Fig. 2. The performance of CSMA/ECA with fixed parameters is compared to CSMA/CA with fixed and dynamic parameters. Simulations results are provided for CSMA/ECA.

the efficiency in (3), because only a fraction of the collisions is avoided. During this transitory phase, the number of stations that successfully transmit (and thus use a deterministic back-off) is a random variable. In the next section, the evolution of this number is modelled as a Markov Chain in order to draw additional conclusions about the transition process.

III. A Dissection of the Convergence Process

Consider a scenario with ς saturated stations and a virtual frame size of V slots, $2 \leq \varsigma \leq V$. We will assume that the transition process occurs in a frame-by-frame basis. Let X_n be the random variable that represents the number of stations that successfully transmitted in the frame n. Then we can model the transition process as a time-homogeneous Markov Chain whose state space is

$$\mathbf{S} = \{S_i | 0 \leq i \leq \varsigma\} \tag{4}$$

As the system runs, it transitions from an initial state S_0 to a (stable) state S_ς.

We are interested in computing the transition probability matrix $\mathbf{P}$ which is the matrix of one step transition probabil-

ities $p_{i,j}$ defined by [3]

$$p_{i,j} = Pr(X_{n+1} = j | X_n = i) \; ; \; 0 \leq i, j \leq \varsigma. \quad (5)$$

Before dealing with the general computation of $p_{i,j}$, we will analyze some results that immediately arise from the definition of the problem and provide some insights about the behaviour of the model. Note that the following properties apply only to the model, and not necessarily to the system that is being modelled. However, they are helpful in computing the transition matrix for the model.

Claim 1: The system is stable when $X_n = \varsigma$, i.e. state S_ς is absorbing.

$$Pr(X_{n+1} = \varsigma | X_n = \varsigma) = 1. \quad (6)$$

Proof: $X_n = \varsigma$ implies that all the stations successfully transmitted in virtual frame n. Therefore, all the stations will deterministically choose the transmission slot in virtual frame $n+1$, specifically they will transmit in the same position in the frame $n+1$ as they did in virtual frame n. As there were no collisions in frame n, there will be no collisions in frame $n+1$. ■

Claim 2: It is not possible that there is one and only one station that randomly selects the transmission slot in a given virtual frame.

$$Pr(X_n = \varsigma - 1) = 0 \; ; \; n > 0. \quad (7)$$

Proof: Seeking a contradiction we assume that there is only one station that randomly selects the transmission slot in virtual frame n. This implies that this station suffered a collision in the previous frame $n-1$. Since a collision occurs when a minimum of two stations transmit in the same slot, there are at least two stations that will randomly select the transmission slot in virtual frame n. This contradicts our assumption. ■

A. Computing the Transition Probability Matrix

After these preliminary results, we face the general problem of computing $p_{i,j}$, i.e. the probability that we have j successful transmissions in the current virtual frame given that there were i successes in the previous frame. There are i stations that deterministically transmit in i different slots, while the rest of the stations $(\varsigma - i)$ randomly transmit in any of the V slots.

Note that for the special case $i = 0$, the problem is reduced to the computation of the number of successes that are obtained when ς stations transmit in V slots and can be solved using the model suggested in [7]. For any other value of i ($i \neq 0$), the approach in [7] is no longer applicable, since it assumes that there are slots reserved for the stations that successfully transmitted in the previous frame. Hence, we are interested in finding another scheme that can be used for any value of i, V and ς.

For large values of V and ς, a brute force approach that sweeps all the different combinations to obtain the transition probability matrix $\mathbf{P}$ is computationally impractical. To compute the first row of the transition matrix, it would be necessary to consider ς stations that could transmit in any of the V available slots, which would account for V^ς possibilities.

[3]Note that we index the rows of the matrix from 0 to ς. This is for consistency with the numbering of the states of the Markov Chain.

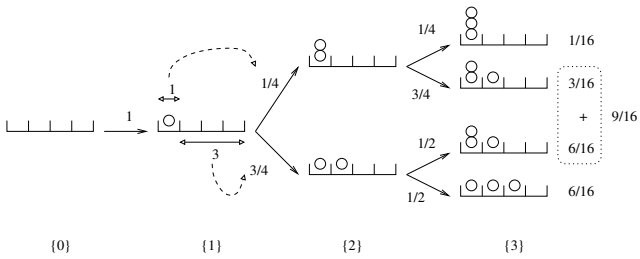


Fig. 3. A tree is used to evaluate the different outcomes that are possible in a system with $\varsigma = 3$ and $V = 4$.

Nevertheless, certain shortcuts are possible to accelerate the computation of $\mathbf{P}$. The reason is that we are interested only in the number of successful slots in a virtual frame, but not in which are those successful slots. In other words, the slots are interchangeable. Similarly, we are not interested in which are the stations that successfully transmitted; all the stations are equivalent from our point of view.

By using the aforementioned interchangeability properties, we propose the following method to compute $\mathbf{P}$. Assume that the previous state is S_0 and we want to compute the probabilities $p_{0,j}$ for all values $0 \leq j \leq \varsigma$. Now consider a transmission in the current frame. This transmission can be in any of the V (for now, empty) slots. Since all these slots are empty, the V possible outcomes are equivalent for our analysis. Each of the V outcomes consists of a slot with one transmission and $V-1$ empty slots. Following the same reasoning, for a second transmission in the same virtual frame, there are only two possible outcomes: *a)* that the transmission slot is the same as the one selected for the first transmission (which occurs with probability $1/V$) or *b)* the two transmissions are in different slots (which occurs with probability $(V-1)/V$). These steps can be repeated to build a tree and obtain all the possible outcomes of interest and the probabilities associated with each outcome. A graphical example is presented in Fig. 3.

In Fig. 3 we show an example for $\varsigma = 3$ and $V = 4$. It is a tree with $\varsigma + 1$ levels. The root represents the $V = 4$ empty slots, and in every level, a new transmission (represented as a ball) is included. The levels are labeled as $\{0\}$, $\{1\}$, $\{2\}$ and $\{3\}$. The edges of the tree are labeled with probability values. At the first level, there is only one node, since the only possible situation (with only one transmission) is one success and three empty slots. Therefore, the edge from the root to the node at the first level is labeled with probability 1. In the transition from level $\{1\}$ to level $\{2\}$ there are two possible options: *a)* that the two transmissions occur in the same slot (with probability $1/4$) and *b)* that the transmissions occur in different slots (with probability $3/4$). This process is iterated until all the transmissions are included, and 4 leafs are obtained. By following the path from the root to the leaf,

the probability of each leaf is computed. The probability that no station successfully transmits can be obtained from the first leaf: From the tree it can be observed that the transition probability from state S_0 to state S_0 (*i.e.* the probability of having zero successes in frame $n+1$ given the fact that there were zero successes in the frame n) is

$$p_{0,0} = Pr(X_{n+1} = 0 | X_n = 0) = \frac{1}{16}. \quad (8)$$

The probability that there is only one success is $p_{0,1} = \frac{3}{16} + \frac{6}{16} = \frac{9}{16}$. The probability of two successes is zero $p_{0,2} = 0$ and the probability of three successes is $p_{0,3} = \frac{6}{16}$. With these values, we have already completed the first row of the transition matrix $\mathbf{P}$. To obtain the values for the second row, one has to assume that there was a successful collision in the previous virtual frame. Therefore, we consider only a subtree of the tree represented in Fig. 3, particularly the one with the root at the node of level $\{1\}$. To compute the third row of the matrix we use as a root the lower node of level $\{2\}$. The last row is computed using only one node, which is the lowest leaf. The transition matrix which is obtained[4] for this example is:

$$\mathbf{P}_{\varsigma=3,V=4} = \begin{pmatrix} \frac{1}{16} & \frac{9}{16} & 0 & \frac{6}{16} \\ \frac{1}{16} & \frac{9}{16} & 0 & \frac{6}{16} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (9)$$

It is not a coincidence that the first two rows of $\mathbf{P}_{\varsigma=3,V=4}$ are the same. Actually, since in level $\{0\}$ all the slots are empty and thus equivalent, there is only one way to place the first ball (transmission). As a consequence, there is always only one node in level $\{1\}$, and the single edge from level $\{0\}$ to level $\{1\}$ takes the value 1.

Claim 3: The first two rows of the transition probability matrix $\mathbf{P}$ are equal.

$$p_{0,j} = p_{1,j} \; ; \; 0 \leq j \leq \varsigma \quad (10)$$

Proof: Consider a tree as the one exemplified in Fig. 3. Then take the subtree with the node of level $\{1\}$ as a root. From this tree, we can obtain the values of the second row (indexed as 1) of the transition matrix $p_{1,j}$. Now, to obtain the first row (indexed as 0), we observe that we use exactly the same tree, but with an additional edge with value 1 and an additional node as a root. Then we can obtain the values of the first row by multiplying the values of the second row by one. ■

We are interested in evaluating how long does it take for the system to leave the transitory phase and begin the collision-free operation. We consider an initial state S_0 in which all the stations randomly choose their transmission slot and then we use the transition matrix $\mathbf{P}$ to evaluate the marginal distributions in subsequent frames. Let

$$\boldsymbol{\pi}_n = \{Pr(X_n = i), 0 \leq i \leq \varsigma\} \quad (11)$$

[4]A script in Octave to compute the transition matrix for any value of V and ς is available upon request.

be the vector of the marginal probabilities at stage n, and $\boldsymbol{\pi}_0 = [1, 0, ..., 0]$ the initial vector. This means that the initial state is S_0 with probability 1. Then the vector $\boldsymbol{\pi_n}$ can be obtained by:

$$\boldsymbol{\pi}_n = \boldsymbol{\pi}_0 \mathbf{P}^n. \quad (12)$$

The last component of vector $\boldsymbol{\pi}_n$ is precisely the value of interest for our study $Pr(X_n = \varsigma)$, which is the probability that the system has reached the stable collision-free state. One particularity of our evaluation of the transition curve is that we have considered that the transition step contains $2 * V$ slots *i.e.* two virtual frames. This is an approximation of the expected backoff of those stations that suffered a collision. We are implicitly assuming that the probability that the same station suffers multiple successive collisions is low, which is true for low values of ς. As the value of ς approaches the value of V, the assumption is no longer valid.

B. Validation by Simulation

The model presented above is based on two approximations with respect to the actual CSMA/ECA operation. The first one is that, in the model, the convergence process occurs in a frame-by-frame basis. In contrast, the CSMA/ECA algorithm allows that the same station re-attempts transmission (and eventually succeeds) in the same virtual frame. Actually, the virtual frame concept is not intrinsic of CSMA/ECA and it is an abstraction we have used for the analysis. The second concession to simplicity is that the exponential growing of the contention window has been neglected in our model. As a consequence of these two concessions (frame-by-evolution and static contention window), our model provides only an approximation to the expected behaviour of CSMA/ECA.

At this point it should be clear that, being the transitory operation a random convergence process, only probabilistic guarantees can be offered regarding its duration. Fig. 4 plots the probability that the system has reached the collision-free operation in a given slot. The results obtained from the model are compared to those obtained from simulation. It can be observed that the transition process is slower for higher values of ς.

C. Disruption of the Stationary Operation

Although the system is expected to run in the collision-free mode of operation for most of the time, there are two events that can disrupt the stationary operation: a channel error and a new entrant. The model can be used to assess the recovery curves associated with these events. It is necessary to force the initial state to $S_{\varsigma-1}$. Regardless of the fact that the system will never transition to $S_{\varsigma-1}$, it is possible to use it as an initial state. It precisely reflects the fact that all stations but one are using a deterministic backoff. The initial vector under consideration is: $\boldsymbol{\pi}_0^D = [0, ..., 0, 1, 0]$.

And the marginal probabilities of subsequent steps:

$$\boldsymbol{\pi}_n^D = \boldsymbol{\pi}_0^D \mathbf{P}^n. \quad (13)$$

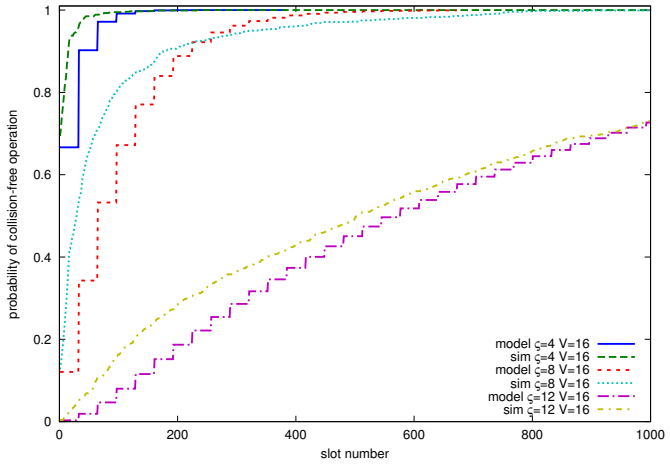


Fig. 4. The transition curves obtained using the model and simulation are compared for a value of $V = 16$ and various values of ς.

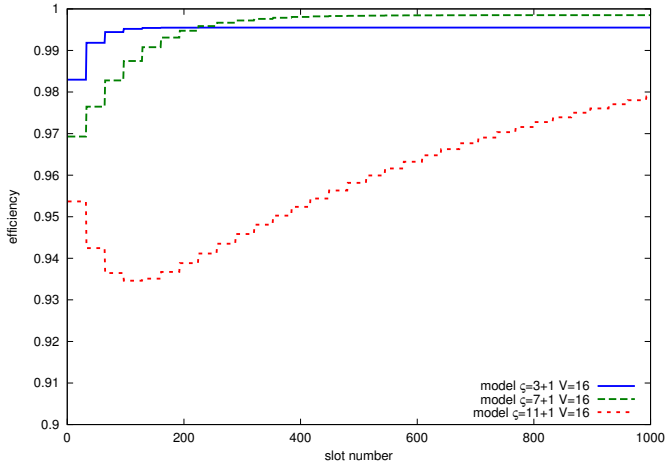


Fig. 5. Recovery curves after a channel error or new entrant.

Provided that current state is S_i, we use the maximum number of collisions (worst case) in the previous step as an approximation of the actual number of collisions in the previous step:

$$\kappa_i \approx \lfloor \frac{\varsigma - i}{2} \rfloor. \tag{14}$$

where $\lfloor \cdot \rfloor$ is the floor operator. Then, using the approximation $T_c \approx T_s$, the efficiency of the system in the step $n-1$ is:

$$\phi_{n-1} \approx \sum_{i=0}^{\varsigma} \frac{2 \cdot i \cdot T_s}{(2 \cdot i + \kappa_i) \cdot T_s + (2 \cdot V - 2 \cdot i - \kappa_i) \cdot T_e} \pi_n^D(i), \tag{15}$$

where the expectation of the backoff of those stations that suffer collisions is considered to be twice as much as V.

Fig. 5 shows the recovery curves obtained from (13)-(15). The transitory phase associated with new incorporations to the contention can be avoided by means of Smart Entry, which will be described in Subsection IV-B.

IV. Implementation Issues

In this section we address the coexistence of CSMA/ECA with the legacy protocol. We also propose a rational way to join the contention and study the performance of the system when we release the ideal channel assumption.

A. Coexistence with legacy CSMA/CA

A promising field of application of the proposed CSMA/ECA is the successful protocol suite IEEE 802.11. Nevertheless, given the large number of deployed networks and terminals, any new version of the medium access control algorithm should be backward compatible with the already existing equipment. Furthermore, to guarantee the smooth coexistence of new and legacy stations, those stations running CSMA/ECA should consume a fair amount of the available bandwith.

The only difference between CSMA/CA and CSMA/ECA as presented in Algorithm 1 can be found in line 11. CSMA/CA randomly chooses the backoff value from the minimum contention window ($b \leftarrow \mathcal{U}[0, CW_{min} - 1]$), while CSMA/ECA deterministically chooses as a value the size of the virtual frame ($b \leftarrow V$). In order to fairly compete with legacy stations, it is desired that

$$V = \lceil E\left[\mathcal{U}[0, CW_{min} - 1]\right] \rceil, \tag{16}$$

where $E[\cdot]$ represents the expectation operator and $\lceil \cdot \rceil$ is the ceiling operator. This selection of the virtual frame size guarantees that the expected number of slots that a station waits after a successful transmissions is approximately the same for both CSMA/CA and CSMA/ECA.

To validate this idea, we performed simulations for a scenario in which half of of the stations run CSMA/CA while the other half use CSMA/ECA. The values chosen for the MAC parameters are $CW_{min} = 32$ and $V = 16$. The rest of the parameters are taken from the IEEE 802.11b specification. Each simulation runs for 10000 slots and each scenario is repeated ten times. The number of competing stations range from two to forty (only even values are considered). When a value of 40 stations is indicated, it actually means 20 CSMA/ECA stations plus 20 CSMA/CA stations.

The plot in Fig. 6 presents the results in three different curves. The first curve, marked with boxes, shows the overall channel efficiency. The second curve, marked using circles, is the amount of channel time devoted to successful transmissions of the CSMA/ECA stations. Finally, the third curve is marked using triangles and is the amount of channel time devoted to successful transmissions of CSMA/CA stations.

It can be observed that CSMA/ECA flows obtain higher channel utilization than CSMA/CA flows thanks to the reduced collision probability. This small advantage can be seen as an incentive for legacy stations to shift to CSMA/ECA for the greater benefit of the network. Jain's index [8] can be used to assess the fairness of the system:

$$fairness = \frac{(\phi_{csma/eca} + \phi_{csma/ca})^2}{2(\phi_{csma/eca}^2 + \phi_{csma/ca}^2)}. \tag{17}$$

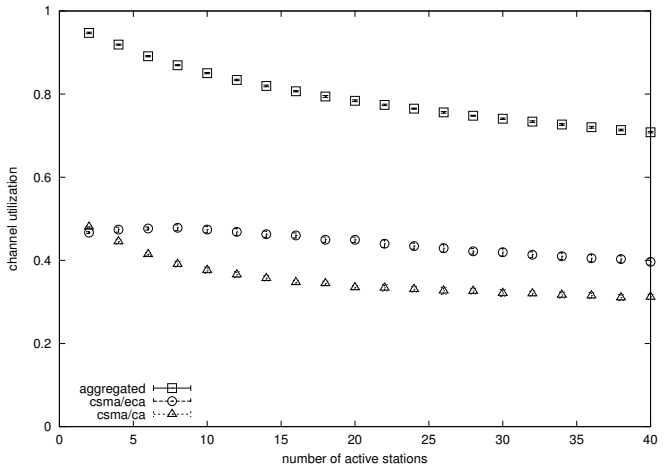


Fig. 6. Half of the stations run CSMA/ECA, while the other half run CSMA/CA. The figure shows the channel utilization achieved by each group.

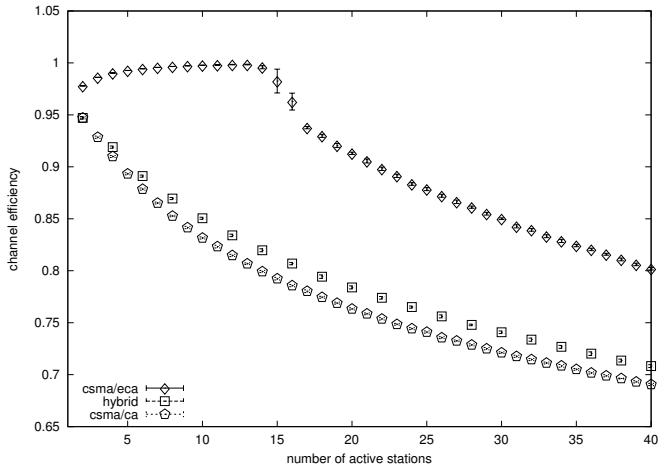


Fig. 7. The channel efficiency obtained for pure CSMA/ECA and pure CSMA/CA scenarios. The efficiency in a hybrid (mixed) scenario is also included.

The possible outcomes range from 0.5 (worst case) to 1 (best case). We obtained results higher than 0.98 when comparing the channel utilization of CSMA/ECA and CSMA/CA in a mixed scenario.

The benefits of using CSMA/ECA are greatly diminished in the presence of legacy stations since the collision-free operation is never reached. Nevertheless, a network running a mixture of CSMA/CA and CSMA/ECA stations will offer equal or better performance that a pure CSMA/CA network, since some of the collisions will be avoided.

To assess the benefits of using CSMA/ECA we simulate three different scenarios, namely, a pure CSMA/ECA, a mixed CSMA/ECA and CSMA/CA and a pure CSMA/CA. The results are compared in Fig. 7. Note that there is a curve (marked with squared boxes) that is common in Fig. 6 and in Fig. 7.

It can be observed that, thanks to the enhanced collision avoidance mechanism, a larger fraction of the channel time is devoted to successful transmissions when only CSMA/ECA is used. For a number of active stations up to the size of the virtual frame size V, the efficiency is almost 1.

B. Smart Entry

So far we have assumed that the number of contenders is fixed. Nevertheless, in a real network, the stations join an leave the contention depending on the load that they receive from the upper layers of the protocol stack.

Ideally, the system will run in the collision-free stable mode of operation. At this point, if a station that joins the contention selects the first transmission slot randomly, it poses the collision-free mode of operation of the system at risk: it may provoke a collision and move the system back to its transitory (collision-prone) mode of operation. To avoid this situation, the stations that are not actively contending for the channel should keep track of the empty slots in each virtual frame. When one of those stations receives a packet from the upper layer, it already knows which slots are expected to be empty, and can schedule the first transmission accordingly.

If Smart Entry is to be used, the first line of Algorithm 1 has to be substituted by Algorithm 2. It includes an array called `slotNumber[]` to keep track of the status of each slot of the frame. The size of this array is precisely the size of the virtual frame V. With the modification presented in Algorithm 2, a station joining the contention transmits in the first empty slot.

Note that while the station is delaying the first transmission attempt, it marks the positions in the array as free. This behaviour prevents a deadlock in the case in which all the slots are busy. If there are no free slots, the station will delay its transmission attempt V slots, and then deliberately prompt a collision in order to free some slots for a future transmission attempt.

C. Releasing the Ideal Channel assumption

Stations cannot distinguish channel errors from collisions, and therefore use a random backoff after all packet losses. We want to stress the proposed protocol by introducing packet errors with probability of 10^{-2}. This 1% threshold was used as a standard measure of robustness by the IEEE 802.11 committee [9].

The simulations in Fig. 8 are performed in the presence of imperfect channel conditions. The packet errors are treated as collisions by the stations and, hence, interfere in the enhanced collision avoidance mechanism. It can be observed that CSMA/ECA clearly outperforms CSMA/CA in lossy channels.

V. Related Work

In [5] it is shown that there is a fundamental limit on the efficiency of completely random access protocols, in which the transmission slot is chosen without using any prior information. Then, it is explained that CSMA/ECA can overcome that limit by using a random behaviour after failures (to trigger

```
   /* Initialize slotNumber[]                    */
1  for i ← 0 to V − 1 do
2      slotNumber[i] ← unknown ;
3  end
4  i ← 0 ;
   /* Scan the channel while waiting for a
      packet from the upper layers.              */
5  while True do
6      if there is a packet ready to transmit then
7          if slotNumber[i] is free then
8              transmit ;
               /* Leave Smart Entry and move
                  to normal CSMA/ECA
                  operation.                     */
9              break ;
10         else
11             wait 1 slot ;
12             slotNumber[i] ← free ;
13         end
14     else
15         wait 1 slot ;
16         if channel sensed busy then
17             slotNumber[i] ← busy ;
18         else
19             slotNumber[i] ← free ;
20         end
21     end
22     i ← (i + 1) (mod V) ;
23 end
```

Algorithm 2: Smart Entry

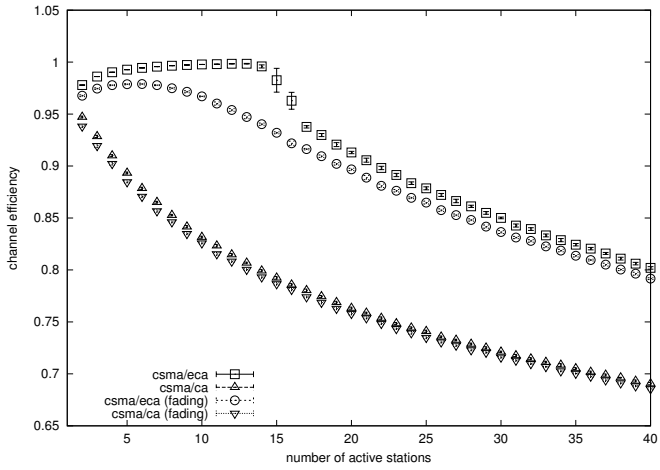


Fig. 8. The channel efficiency delivered by CSMA/ECA and CSMA/CA in an unreliable channel.

a change) and a deterministic behaviour after successes (to stabilize the system).

In [10], simulations are used to assess the performance of CSMA/ECA in saturated, non-saturated and hybrid (a combination of saturated and non-saturated) scenarios. CSMA/ECA is shown to perform equal or better than CSMA/CA in all the considered scenarios. Specifically, the two protocols deliver the same throughput in those scenarios in which the network is able to absorb all the offered traffic. However, when the traffic load overwhelms the network, CSMA/ECA performs better than CSMA/CA. Traffic prioritization in CSMA/ECA is addressed in [11]. An analytical model to capture the behaviour of CSMA/ECA in stationary operation for both rigid and elastic flows is presented in [12].

VI. Conclusions

In this article we address the problem of collisions in CSMA networks. Our finding is that, instead of using a random backoff after all transmission attempts, it is better to use a random one after collisions and a deterministic one after successes. It reduces the chances of collisions as soon as two or more stations successfully transmit. As the system runs, it progressively converges to a collision-free operation that considerably improves the channel efficiency.

The proposed protocol outperforms CSMA/CA and, in the most typical scenarios, it even surpasses the theoretical upper bound associated with CSMA/CA networks that allow for dynamic parameter adjustment. Additionally, CSMA/ECA does not add any additional complexity to the implementation, it can fairly coexist with already deployed networks and it is robust against unreliable channel conditions.

Acknowledgments

The authors benefited from the discussion with A. Banchs, B. Bellalta, A. Lozano and A. Vinel. This work was partially supported by the Spanish Government under grant TEC2008-06055/TEC.

References

[1] A. Tanenbaum, *Computer Networks*. Prentice Hall PTR, 2002.
[2] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1999 Edition (Revised 2003).
[3] F. Borgonovo and L. Fratta, "A New Technique for Satellite Broadcast Channel Communication," in *Symposium on Data Communications*, 1977, pp. 2.1–2.4.
[4] S. Tasaka, "Stability and Performance of the R-ALOHA Packet Broadcast System," *IEEE Trans. Comput.*, vol. C-32, pp. 717–726, Aug. 1983.
[5] J. Barcelo, B. Bellalta, C. Cano, and M. Oliver, "Learning-BEB: Avoiding Collisions in WLAN," in *Eunice*, 2008.
[6] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, 2000.
[7] T. Liu, J. Silvester, and A. Polydoros, "Performance Evaluation of R-ALOHA in Distributed Packet Radio Networks with Hard Real-Time Communications," in *IEEE VTC*, vol. 2, 1995, pp. 554–558.
[8] R. Jain, *The Art of Computer Systems Performance Analysis*. John Wiley & Sons New York, 1991.
[9] C. Heegard, J. Coffey, S. Gummadi, P. Murphy, R. Provencio, E. Rossin, S. Schrum, M. Shoemake, and T. Inc, "High Performance Wireless Ethernet," *Communications Magazine, IEEE*, vol. 39, no. 11, pp. 64–73, 2001.
[10] J. Barcelo, B. Bellalta, A. Sfairopoulou, C. Cano, and M. Oliver, "CSMA with Enhanced Collision Avoidance: a Performance Assessment," in *IEEE VTC Spring*, 2009.
[11] J. Barcelo, B. Bellalta, C. Cano, A. Sfairopoulou, M. Oliver, and J. Zuidweg, "Traffic Prioritization fo Carrience Sense Multiple Access with Enhanced Collision Avoidance," in *MACOM (IEEE ICC)*, 2009.
[12] J. Barcelo, B. Bellalta, A. Sfairopoulou, C. Cano, and M. Oliver, "Carrier Sense Multiple Access with Enhanced Collision Avoidance: a Performance Analysis," in *ACM IWCMC*, 2009.

PLATAFORMA MULTIFUNCIONAL PARA GESTIÓN DEL CANAL EN REDES IEEE 802.11

Domingo Marrero Marrero, Elsa Mª Macías López, Álvaro Suárez Sarmiento
Departamento de Ingeniería Telemática
Universidad de Las Palmas de Gran Canaria
Campus Universitario de Tafira, Edificio de Electrónica y Telecomunicación, Pab. C, despacho 223
35017 Las Palmas de Gran Canaria
dmarrero@dit.ulpgc.es, emacias@dit.ulpgc.es, asuarez@dit.ulpgc.es

***Resumen*- El uso con garantías de calidad de muchos servicios en redes WLAN (*Wireless Local Area Networks*) que siguen el estándar 802.11 definido por el IEEE no está asegurado a pesar de los esfuerzos realizados por diversos grupos de trabajo, debido entre otros factores, a que no se logran velocidades elevadas, y no se garantizan retardos y *jitter* mínimos, aspectos muy importantes en servicios dependientes del tiempo. Por ello se necesitan incorporar soluciones orientadas a mejorar la QoS (*Quality of Service*). En este artículo se presenta un conjunto de acciones integradas bajo una plataforma multifuncional para mejorar la calidad de redes WLAN basadas en IEEE 802.11. Dos de sus funciones son: 1) mecanismo de regulación de tráfico desde origen y 2) gestión centralizada en los AP (Access Point) para reubicar las estaciones móviles. En la primera, se pretende que los puntos de acceso puedan solicitar a ciertas estaciones limitar su velocidad de transmisión (bits por segundo) a valores calculados dinámicamente, para priorizar unos flujos frente a otros. La segunda acción consiste en que los AP intercambien información de estado y entre ellos se realicen traspasos de estaciones asociadas.**

***Palabras Clave*-**

Redes de área local inalámbricas, IEEE 802.11, QoS, Aplicaciones dependientes del tiempo, Prestaciones.

I. INTRODUCCIÓN

Las tecnologías para redes de área local inalámbricas son cada vez más demandadas por los usuarios, ya sea en soluciones basadas en puntos de acceso [1][2] para acceder a las redes cableadas o a Internet, como soluciones ad-hoc [1][2]. Asimismo, se está avanzando muchísimo en el uso y la estandarización de redes malladas [3] como evolución de las redes ad-hoc. La inexistencia de cables y su ubicuidad, brinda a estas redes, de manera indudable, de una serie de ventajas sobre las redes cableadas, pero por el contrario, aún distan mucho de alcanzar las velocidades y la calidad de éstas. Probablemente, la velocidad no sea en gran parte de los servicios utilizados en la actualidad una prioridad (web informativas estáticas, correo electrónico y transferencia de archivos). En cambio para servicios dependientes del tiempo, como por ejemplo transmisión de video, audio o ambos (VoIP, VoD, radio-internet, video-vigilancia, etc.), la velocidad, pérdida de paquetes, retardo y jitter sí son aspectos determinantes que afectan muchísimo a su calidad y aún no están garantizados en la mayoría de los productos disponibles.

Para mejorar el estándar IEEE 802.11 [1][2][3], en sus diferentes versiones, surgió el estándar 802.11e [4] que permite enriquecer el comportamiento de estas redes modificando el control de acceso al medio mediante la priorización de tráfico y alterando los tiempos de acceso al canal. En esa misma línea se orienta el estándar en desarrollo 802.11n [5]. Aún con esto, el número de investigadores que centran sus esfuerzos en mejorar 802.11e y en proponer otras soluciones o variantes y combinaciones de varias propuestas es bastante elevado [6][7][8]. En cualquiera de los casos, el gran problema que existe, es la limitación del espectro y las regulaciones administrativas existentes (potencia, bandas, canales, etc.).

En esta situación, toda aportación que se introduzca y que redunde en mejorar el comportamiento de las redes inalámbricas en cuanto a la calidad de servicio ofrecida a sus usuarios debe ser considerada, máxime si ellas no representan nuevos protocolos o productos incompatibles y puedan implementarse como valor añadido a los productos existentes.

Es en este contexto donde se presentan dos acciones integradas dentro de lo que hemos llamado plataforma multifuncional para gestión del canal en redes 802.11. La primera de las acciones representa un control dinámico de tráfico centralizado en el AP pero aplicado en las estaciones (en el origen de los datos). La segunda acción que se presenta consiste en que redes donde haya varios AP puedan producirse traspasos de estaciones vinculadas a ellos, obviamente dentro de sus zonas de cobertura comunes. Este traspaso se realizaría en base al cálculo de disponibilidad de canal, número de estaciones, estado del canal, etc.

La estructura del artículo es la siguiente: en el apartado II se realizan unas consideraciones preliminares sobre los problemas existentes en las redes WLAN. En el apartado III presentamos una propuesta de solución integrada de múltiples acciones orientadas a mejorar la percepción del usuario en cuanto a la QoS ofrecida por la red inalámbrica se refiere. En el apartado IV se describe la plataforma de prueba utilizada para analizar las prestaciones y el comportamiento de ambas propuestas, así como un resumen de los resultados obtenidos. Finalmente, se exponen las conclusiones y se

enumeran algunas líneas de trabajo que se están desarrollando.

II. Consideraciones Preliminares

La calidad de servicio en redes de computadores es un aspecto cada vez mas demandado en la actualidad. Muchas soluciones están siendo desarrolladas para aumentar la velocidad, reducir las pérdidas de paquetes, el retardo y jitter. Estas mejoras y soluciones no suelen ser de aplicación general para todas las tecnologías de redes, sino que dependiendo del tipo de red de que se trate, podrían aplicarse o no.

Especialmente en redes locales, al ser redes de difusión, el comportamiento "best effort", no es el más adecuado para garantizar una calidad de servicio determinada. Para ello, soluciones basadas en servicios integrados *IntServ* [9], con implementaciones como protocolos de reserva de recursos o *DiffServ* [10] han sido definidos. Estas soluciones son perfectamente válidas para redes de área local y extensa con interconexiones cableadas, dado que mejoran las prestaciones estableciendo rutas o reservando recursos.

En la figura 1 se ilustra la configuración típica de una red formada como integración de varias redes.

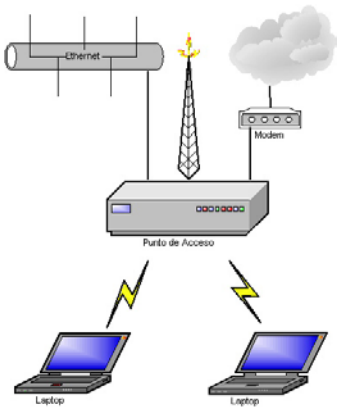


Fig. 1. Esquema general de topologías de redes

En ella podemos distinguir tres zonas diferentes para estudiar los problemas que afectan a la velocidad, retardo y jitter: a) las redes cableadas (generalmente Ethernet) donde protocolos como RSVP (Servicios Diferenciados) y Servicios Integrados intentan mejorar los parámetros de QoS; b) el acceso a la red (Internet) o WAN (ADSL, RDSI, ATM, etc.), en los que se ofrecen diferentes tecnologías, algunas con anchos de banda más garantizados que otras y c) la red de acceso inalámbrico que presenta mayores problemas debido a: interferencias, limitado ancho de banda/velocidad, desconocimiento de estaciones implicadas, proceso de handoff-roaming, etc. Es, en esta red de acceso, donde es más complicado definir soluciones óptimas para garantizar unos valores apropiados para ofrecer QoS a los diferentes servicios. Tanto en modo infraestructura como ad-hoc, se tiene un canal radio compartido con limitaciones de ancho de banda [11] y donde no existen nodos intermedios donde aplicar soluciones, como las aplicadas en redes cableadas.

La mayor parte de las propuestas pasan por alterar el acceso al canal según el tráfico a nivel de MAC, otras utilizando múltiples canales radio, etc. En otras propuestas para topologías modo infraestructura, dado que los puntos de acceso posibilitan el acceso a la red cableada, a Internet o a otras redes, se aplica algún tipo de mecanismo de priorización de paquetes y colas con diferentes prioridades. También se pueden incorporar mecanismos que regulan el ancho de banda de cada conexión dentro del propio AP; con ello evitan un uso abusivo del canal, retrasando las respuestas en las conexiones existentes pero no en el tráfico que se inserta en el canal.

Resumiendo podemos concluir que las mejoras en la QoS se basan en dispositivos que incorporan 802.11e, dispositivos de control de ancho de banda o limitación de tráfico en el AP.

III. Descripción general de la plataforma propuesta

En este artículo se presenta una propuesta que ayuda a mejorar la calidad de servicios en redes inalámbricas 802.11 mediante una distribución del uso de los diferentes recursos (canales, AP, Ancho de Banda, etc.). Para ello se pretende dotar a las estaciones y AP participantes de las aplicaciones necesarias para dar soporte a una plataforma multifuncional, modular y multicapa que integra múltiples líneas de actuación. Nuestra idea parte de tres premisas:

1. Los AP ocupan una posición estratégica y clave en el control del funcionamiento de la red
2. Son la puerta de acceso a la red cableada, y
3. Tienen o deben tener un conocimiento detallado de estaciones, capacidades y servicios activos o disponibles

Por todo lo anterior, creemos que los AP deben tener un mayor protagonismo en el control y gestión de la red y sus limitados recursos (especialmente el uso del canal radio).

Esta plataforma se podría materializar con una aplicación a nivel de sistema o formando parte del sistema operativo en cada estación, que interactúe con los AP. A este proceso ubicado en las estaciones inalámbricas lo denominaremos "*agente*". El seguimiento y control sería gestionado y regulado desde el AP mediante un proceso de control al que hemos denominado proceso "*gestor"*. Los agentes en cada estación deberán estar en comunicación directa y constante con el gestor del AP al que esté asociado.

En la figura 2 se esquematiza el modelo basado en un gestor en cada AP y los agentes en cada estación.

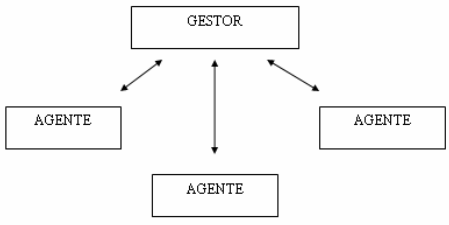


Fig. 2. Modelo Gestor-Agente

A continuación describimos dos de las funcionalidades de esta plataforma:

- Módulo I: Regulación del Tráfico
- Módulo II: Traspaso de Clientes

Módulo I: Regulación del Tráfico

Como primera acción dentro de la plataforma, proponemos que los AP puedan forzar a las estaciones que comparten el canal a que limiten la tasa de bits por segundo del tráfico inyectado en la red, si este es menos prioritario que otro/s existente/s. Con un ejemplo muy simple se entiende la idea. Si dos equipos portátiles se conectan a un AP y uno de ellos va a establecer una videoconferencia y el otro una transferencia de un archivo, el AP podría forzar al segundo a limitar su tasa de bits por segundo hasta unos valores que afecten en la menor medida posible al primero. Esto requiere de un control de admisión de estaciones [12] más elaborado para saber a priori qué servicios o conexiones van a ser establecidas por cada estación y una necesaria clasificación del tráfico y priorización posterior.

Esta idea, iniciada con un control del tráfico en base al tipo de acceso [12][13], se materializa en la regulación del tráfico desde la fuente que genera dicho tráfico por indicación o exigencia desde el AP. Cada AP, con el conocimiento que deberá tener, tanto de las conexiones entrantes como de las existentes, calculará de manera dinámica la cantidad de tráfico (bits por segundo) que cada estación debería insertar para sus conexiones. Con ello, el tráfico no dependiente del tiempo será reducido en beneficio de los que sí lo sean. Para garantizar que cada estación emita a la tasa indicada, los AP incorporarían un *sniffer* [14] que detecte, calcule y compruebe los mismos. En caso de superarse alguno de ellos, las conexiones correspondientes pueden ser bloqueadas.

Como primer paso, dado que es necesario que los AP conozcan qué conexión se va a establecer, se plantea que a nivel de aplicación exista un *control de admisión* en cada AP, no sólo como control de usuarios sino en nuestro caso, de manera especial, qué servicio se va a usar y qué requerimientos tienen de ancho de banda o velocidad. Con ello, cada AP podrá calcular y distribuir el ancho de banda disponible del canal que gestiona entre las conexiones existentes. Si los requerimientos de ancho de banda de las conexiones solicitadas no se garantizaran, sería potestad de cada cliente decidir si continúa o no con la conexión con las características que el AP le otorgue. Caso de aceptarse las condiciones, será preceptivo permitir que el AP tome cierto control a través del agente, del tráfico inyectado por cada estación. El agente, de forma dinámica, podrá recibir las indicaciones del AP y aplicar las modificaciones de velocidad que le fuesen requeridas.

En la figura 3 se ilustran los diferentes componentes del módulo descrito.

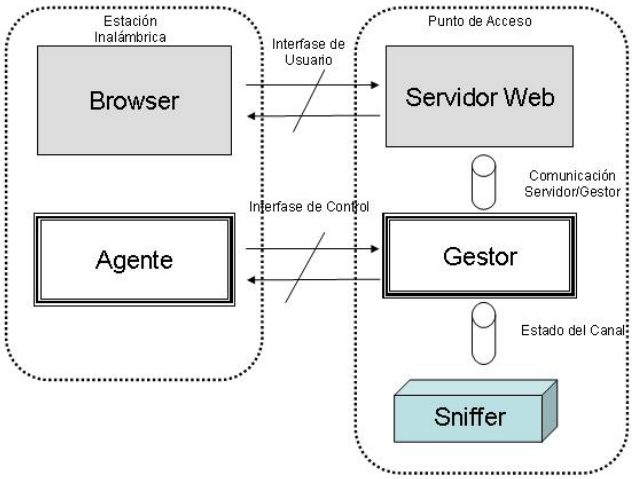


Fig. 3. Componentes para la regulación del tráfico

El cliente mediante su browser interactúa con el servidor web del AP que da acceso a la red (control de admisión) y solicita una conexión (servicio y/o requerimientos de velocidad). El gestor, mediante los correspondientes comandos, indicaría las restricciones o modificaciones (aumentos o reducciones) de velocidad a los agentes, caso de existir éstas. Si se sobrepasaran los límites impuestos o si el cliente no respondiera a los comandos de control, el AP bloqueará las conexiones existentes o no permitirá el acceso. Para la regulación del tráfico, los agentes reciben del gestor los siguientes comandos:

1. *Hello* para detectar la presencia de los agentes y poder recabar, en la respuesta, información de estado de cada estación, por si fuese necesaria para el gestor. El envío de este comando se realiza a modo de *polling* temporizado (varios segundos) sobre cada agente participante.
2. *Establecer* o *Eliminar* determinada regulación de tráfico. El primer comando se usa para indicar a la estación correspondiente la máxima tasa de bits por segundo puede inyectar. El segundo comando elimina la restricción impuesta previa, si existiese.
3. *Fin de tiempo de conexión permitida* controla la duración de la conexión, como complemento necesario al control de admisión de usuario/servicio.

Módulo II: Traspaso de Clientes

En la figura 4 se ilustra una situación de redes en modo infraestructura muy común que da pie a la propuesta planteada con este módulo.

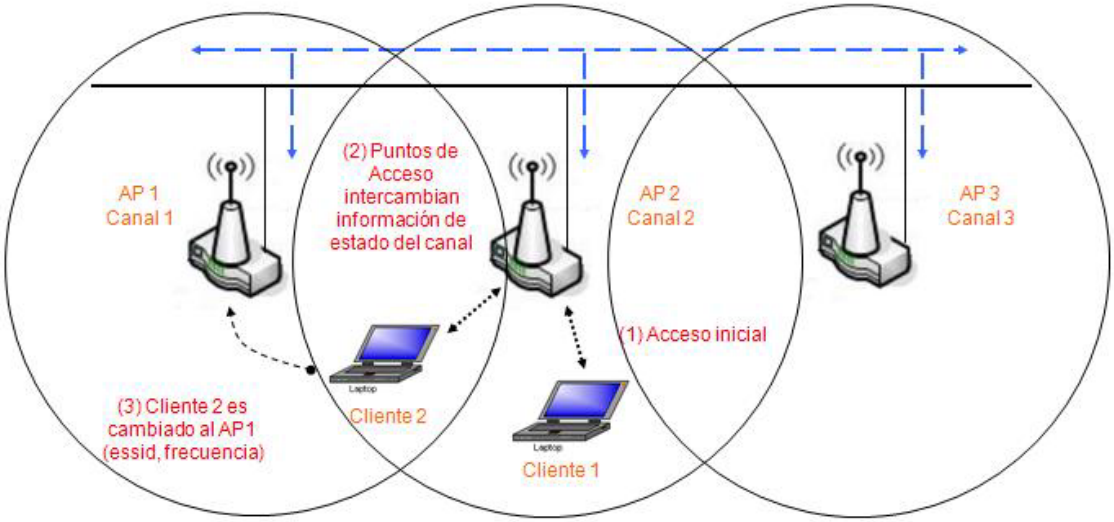


Fig. 4. Estación en cobertura de dos o más APs

En ella se puede observar que el cliente 2 detecta dos AP y se asocia al AP2 donde hay otra/s estación/es que pueden estar transmitiendo tráfico o a punto de hacerlo. Para mejorar esta situación, se plantea esta segunda funcionalidad. Consiste en forzar cambios de AP a la/s estación/es seleccionadas entre las asociadas a un AP origen que ofrezcan mejores condiciones (canal libre, mayor ancho de banda disponible, servicios no dependientes del tiempo, etc.) a un AP destino. Esta funcionalidad se restringe a estaciones que detecten más de un AP, como puede suceder en zonas de amplia cobertura (aeropuertos, campus universitarios, etc.) donde se evita que existan zonas oscuras. Con esta idea, se pretende optimizar, en cierta medida, el uso del canal, pues su saturación o sobrecarga puede ser debida a que existen muchas estaciones en la intersección de dos canales o celdas. Si este caso se presenta, pudo ser debido simplemente porque el criterio para la elección del AP2 por parte del cliente 2 a la hora de asociarse a uno de ellos, se realizó simplemente por cuestiones de mejor calidad de señal (generalmente relacionado con la distancia) y no por información de estado del canal.

Si analizamos mejor la figura 4, vemos que los clientes 1 y 2 están asociados al AP2 o intentando hacerlo, e incluso transfiriendo datos entre ellos o cualquier otra estación accesible. Vemos que el canal 1 gestionado por el AP1 está libre y en cambio en el canal 2 está siendo compartido por las comunicaciones del cliente 1 y 2. Si el cliente 2 estuviera asociado al AP2 y se cambiara al AP1 la calidad/velocidad de sus comunicaciones sería mejor que en el canal 2, así como la disponibilidad de canal para el cliente 1.

Por tanto, no siempre la elección automática del AP es la óptima pues se pueden detectar niveles de señal similares, pero el AP1 en el canal 1 no tenga estaciones conectadas y en cambio el AP2 tenga un mayor número de potenciales clientes.

Para implementar esta segunda funcionalidad, vamos a hacer uso de los gestores que conocen las estaciones que tienen asociadas, y entre ellos se intercambien esa información por la red cableada a través del sistema de distribución. Con la información recibida, cada AP, a través de su gestor, podría ofrecerse como potencial receptor de algún cliente de otro AP, liberando con ello a este último de un cliente o varios. Ante un determinado ofrecimiento, el AP sobrecargado podría iniciar un proceso que hemos denominado "**de cambio de AP o traspaso**" para una estación cliente conectada. La decisión de qué estación se traspasaría, se realizaría por el conocimiento de todas las existentes y los servicios utilizados. Podría hacerse el traspaso de aquella que esté en la intersección o en el caso de que hubiera varias, aquella que al traspasarse libere tráfico del canal o se mejore la suya en la nueva red con infraestructura. El gestor de cada AP, al recibir el mensaje de cambio de AP o traspaso analiza su situación actual y determina el proceso de traspaso y a qué estación se aplicaría. Para ello primero notifica al AP receptor que se ha de iniciar dicho proceso y luego tras la comunicación y confirmación por el agente de la estación correspondiente, este se reasocia al nuevo AP.

Para explicar mejor esta funcionalidad se presenta a continuación la secuencia de acciones que tiene lugar entre dos puntos de acceso y la estación inalámbrica que se traspasa:

1. Los gestores difunden (broadcasting) por la red cableada información de estado (estaciones, servicios, características, etc.).
2. Si el AP1 está en mejores condiciones que el AP2 (comparando información de estado recibida desde el AP2) responde de forma unicast al AP2 ofreciéndose a recibir algún cliente.
3. El AP2 al recibir el ofrecimiento del AP1, consulta a todos los clientes si detectan al AP1. Con los datos de aquellas estaciones que respondan afirmativamente se procesará cual está en peores condiciones para el servicio que está utilizando, y se decide cuál traspasar.

4. Si la estación inalámbrica 2 responde que sí ve al AP1, el gestor del AP2 podría forzar por medio del agente correspondiente a que el cliente especificado se traspase al AP1. Esta estación cliente deberá confirmar que inicia el proceso de traspaso.
5. Una vez está asociado al AP1, este último deberá confirmar al AP2 que el cliente está asociado. Si no, se perdería la conexión con el agente desde cualquiera de los dos AP implicados. Además esto garantiza que se realice perfectamente la baja y el alta en los registros correspondientes de cada gestor.
6. Finalmente el cliente inalámbrico 1, sin moverse, estará asociado al AP1. En cualquiera de los casos, el usuario de esta estación deberá ser notificado de cualquier incidencia y en caso de situación irregular, podrá reiniciar la asociación en el AP que desee.

Los comandos vinculados con cada acción son:

1. Difusión de estado entre AP (broadcast)
2. Respuesta unicast al AP con ofrecimiento
3. Indicación de traspaso desde AP a estación
4. Confirmación de inicio de traspaso desde estación a AP
5. Confirmación de traspaso desde AP destino a AP origen

Los comandos 1, 2 y 5 son intercambiados entre AP y los comandos 3 y 4 entre AP y estación.

Con la secuencia de acciones descrita anteriormente y usando los comandos presentados, los gestores podrán realizar el traspaso de clientes, con la necesaria conformidad de los agentes. Esto es necesario y preceptivo para evitar lo que denominamos estación cliente "aislada", que se daría cuando el proceso de traspaso no se culmina correctamente y la estación seleccionada no quede asociada a ningún AP. Si el agente de la estación que se traspasa no recibe ningún mensaje del nuevo AP, activaría un proceso de scanning y se asociaría a cualquiera de ellos, iniciándose el proceso completo.

Con esta nueva funcionalidad, las estaciones que detectan varios AP, sin necesidad de moverse físicamente, se asociarían al AP más óptimo de forma dinámica y dirigida por los gestores de forma distribuida. Los gestores llevarán un registro de estaciones y las altas y bajas correspondientes vinculadas a los traspasos. Además, algo que consideramos muy importante, sin intervención de los usuarios (de forma transparente).

IV. Plataforma de Test y Resultados

Para evaluar la plataforma propuesta, hemos implementado una solución basada en Ordenadores Personales que ejecutan el sistema operativo Linux. Dos de estos ordenadores actúan como AP y se les ha habilitado la funcionalidad de router Linux [15] para implantar el gestor. Por otro lado, en cada estación asociada al punto de acceso se instalan los agentes. Ambos programas, gestor y agentes, han sido programados en el lenguaje C.

Resultados obtenidos para el módulo I: Regulación de Tráfico

En la figura 5 se muestra la plataforma de test utilizada para el módulo I. En las pruebas que se realizaron, a cada estación sólo se le ha permitido un servicio o conexión. En versiones posteriores se habilitarán varios servicios por estación. En este caso será necesario definir un método de distribución de ancho de banda más elaborado por cada AP. Las características de los dispositivos utilizados se muestran en la tabla 1.

Estación / hub	*Hardware*	*NIC*
Estación cableada 1 (Linux Fedora Core 3 operating system)	PC, Pentium III 1Ghz, 512K RAM[1]	Ethernet /Fast Ethernet 10/100BaseT
Estación cableada 2 (Linux Fedora Core 2)	PC, Pentium II, 400Mhz, 128M RAM	Ethernet /Fast Ethernet 10/100BaseT
Estación inalámbrica 1	Pentium III 1Ghz, 256M RAM	*PCMCIA*[2] Compaq [16] IEEE 802.11b
Estación inalámbrica 2	Pentium IV 3Ghz, 1G RAM	PCMCIA Dlink[17] IEEE 802.11b/g
Linux router (Fedora Core III)	PC, Pentium II 400Mhz, 196M RAM	PCMCIA Compaq IEEE 802.11b
Hub	Genius 8 Ports	Ethernet /Fast Ethernet 10/100 Mbps

Tabla 1. Características técnicas del hardware y software utilizado de los equipos mostrados en la figura 5

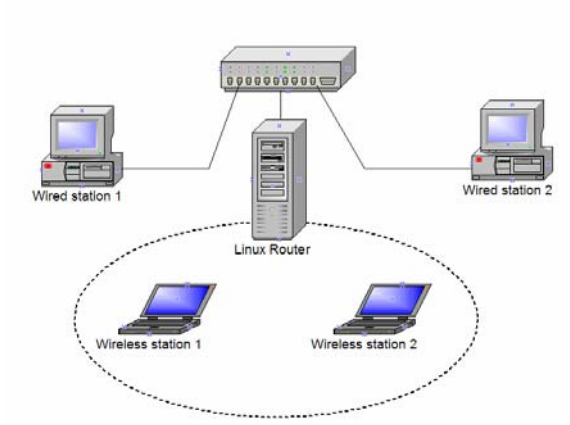


Fig. 5. Plataforma utilizada para las pruebas experimentales del módulo I

Para acceder a la plataforma multifuncional, el gestor incorpora un portal cautivo. Cada usuario deberá identificarse y especificar qué servicio va a utilizar (opcionalmente, notificará sus requerimientos de velocidad). Con ello, como ya se comentó, el gestor podrá calcular qué regulación debería aplicarse a la estación entrante y/o resto de estaciones ya asociadas.

En la figura 6 se muestra la interfaz gráfica de la página de entrada al portal cautivo.

Fig. 6. Interfaz web del portal cautivo del gestor

Una vez aceptada la conexión, se notifica al agente y al usuario las restricciones de velocidad que tendrá. Los agentes procesan los comandos del gestor de la siguiente forma:

- Ante el comando *hello*, deberá responder automáticamente con información de estado del cliente correspondiente.
- Ante los comandos *establecer* y *eliminar regulación*, se utilizará la orden apropiada del kernel para reducir/aumentar la tasa de bits por segundo y posteriormente, confirmar al gestor su aplicación. Esto se realiza haciendo uso de la utilidad tc (traffic control) [18] de Linux.
- Ante el comando *fin de tiempo de conexión permitida* deberá confirmar al gestor este hecho y notificarlo al usuario. Tras este comando, el servicio que estaba siendo utilizado debería estar bloqueado por haberse excedido el tiempo de conexión.

En la figura 7 se ilustra, de forma muy general, las acciones que realiza el gestor (parte izquierda de la figura) y el agente (parte derecha).

Hay que tener en cuenta, que dado el dinamismo de estaciones que entran o salen de la red, se requiere adaptar el tráfico no sólo para la estación que se asocia, sino también para el resto de estaciones. Por ello existe en el gestor una base de datos con todas las estaciones participantes para controlar su estado y sus regulaciones de tráfico. De forma repetitiva y basada en temporización, cada cierto tiempo (configurable para no sobrecargar con tráfico de control la red) se sondea a todas las estaciones para determinar su estado.

La comunicación se ha realizado mediante el protocolo de transporte UDP. Esto permite no sobrecargar el canal con mucho tráfico y reducir las conexiones existentes en los AP (recursos limitados). Además, el protocolo de comunicación y las acciones a realizar siguen un modelo petición-respuesta.

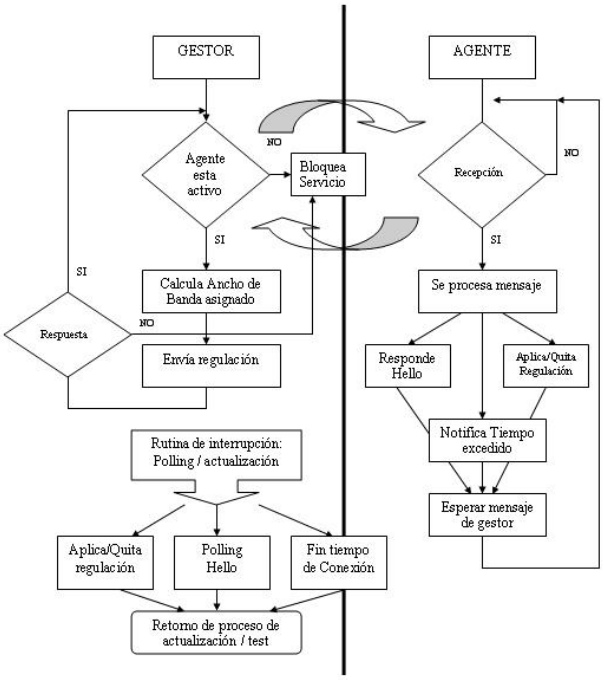


Fig. 7. Organigrama del funcionamiento Gestor-Agente en el módulo I

Como prueba inicial hemos establecido dos comunicaciones sin regulación de tráfico, una entre la estación 1 inalámbrica con la estación 1 en la red cableada, y otra comunicación entre la estación 2 inalámbrica con la estación 2 en la red cableada. Para forzar un gran volumen de tráfico, hemos utilizado la aplicación iperf [19]. Un resumen de los datos obtenidos se muestra en la tabla 2.

Fuente-> Destino	***Tasa***	***Información***
Estación inalámbrica 1 → Estación cableada 1	1.73 Mbps	2.08 Mbytes
	1.75 Mbps	2.11 Mbytes
	1.73 Mbps	2.08 Mbytes
Estación inalámbrica 2 → Estación cableada 2	3.66 Mbps	4.39 Mbytes
	3.69 Mbps	4.44 Mbytes
	3.68 Mbps	4.43 Mbytes

Tabla 2. Máximas velocidades alcanzadas sin regulación

Para evaluar el mecanismo de regulación, el gestor fuerza a que la estación inalámbrica 1 no supere los 125 kilobits por segundo. Los resultados se muestran en la tabla 3. Nótese que, como era de esperar, la comunicación entre las otras estaciones obtiene mejores resultados dada la mayor disponibilidad de ancho de banda

Fuente -> Destino	***Tasa***	***Información***
Estación inalámbrica 1 → Estación cableada 1	125 Kbps	168 Kbytes
	125 Kbps	168 Kbytes
	125 Kbps	168 Kbytes
	125 Kbps	168 Kbytes
	125 Kbps	168 Kbytes
Estación inalámbrica 2 → Estación cableada 2	5.27 Mbps	6.32 Mbytes
	5.24 Mbps	6.28 Mbytes
	5.26 Mbps	6.30 Mbytes
	5.31 Mbps	6.38 Mbytes
	5.27 Mbps	6.33 Mbytes

Tabla 3. Máximas velocidades alcanzadas con regulación

También realizamos una transferencia de datos entre la estaciones 1 usando FTP (tráfico no prioritario) y abrimos una sesión RTSP/RTP entre las estaciones 2 (tráfico prioritario) usando la aplicación VideoLAN [20]. Los resultados obtenidos demuestran que, sin nuestro mecanismo de regulación de tráfico, se observa una reproducción del video intermitente mientras que al usar la regulación solventamos este problema mejorando la percepción final del usuario.

A la vista de las diferentes pruebas obtenidas, se concluye que ante un ancho de banda limitado, aplicar técnicas de regulación de tráfico entre las estaciones involucradas mejora los parámetros de retardo, sincronismo, etc.

Resultados obtenidos para el módulo II: Traspaso de Clientes

Para evaluar la implementación del módulo II se ha utilizado la plataforma de test mostrada en la figura 8.

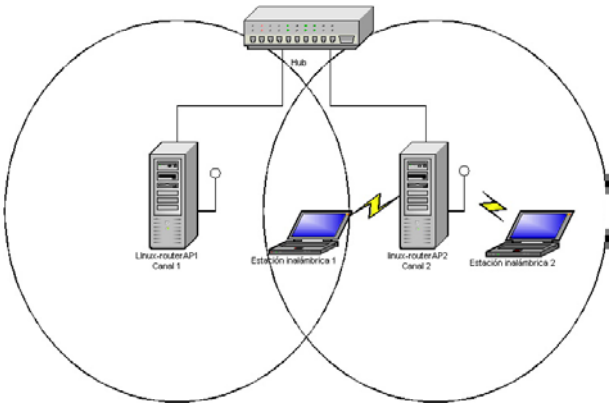


Fig. 8. Plataforma utilizada para las pruebas experimentales del módulo II

En la tabla 4 se detallan las características de los equipos utilizados para evaluar la funcionalidad planteada.

Estación / hub	*Hardware*	*NIC*
Estación inalámbrica 1	Pentium IV 3Ghz, 1G RAM	PCMCIA Dlink[17] IEEE 802.11b/g
Estación inalámbrica 2	Pentium III 1Ghz, 256M RAM	*PCMCIA*[2] Compaq [16] IEEE 802.11b
Linux router1 (Fedora Core III)	PC, Pentium II 400Mhz, 196M RAM	PCMCIA Compaq IEEE 802.11b
Linux router2 (Fedora Core III)	PC, Pentium II 400Mhz, 128M RAM	PCMCIA Compaq IEEE 802.11b
Hub	Genius 8 Ports	Ethernet /Fast Ethernet 10/100 Mbps

Tabla 4. Características técnicas del hardware y software utilizado de los equipos mostrados en la figura 8

Tras varias pruebas realizadas, se ha comprobado que con el uso de esta funcionalidad, la estación cliente 1 aparece asociada al AP1 dejando el canal 2 completamente libre para la estación cliente 2. En esta plataforma, dado que solo la estación 1 ve a ambos AP, es la única que podría ser objeto de traspaso. Hemos hecho pruebas con dos estaciones en la intersección, y en este caso, el proceso es mucho más elaborado, dado que el AP2 determinará dinámicamente que estación traspasar en base al servicio que esté utilizando cada cliente y el conocimiento que tiene del estado del AP receptor. En unos casos traspasaba una estación y en otros casos a otra, en función de las condiciones en origen y en destino (conexiones, número de estaciones, etc.). Esto nos confirma la flexibilidad del proceso y su dependencia de las condiciones.

Para tener un mayor control de estas acciones, las características para realizar el traspaso son configurables en el módulo desarrollado (umbrales para comparación, número de clientes, servicios soportados, umbrales de traspaso, condiciones temporales, etc.).

Los comandos (PDU) transmitidos entre todos los elementos participantes se soportan sobre el protocolo de transporte UDP, para no sobrecargar la red. Para evitar bloqueos ante mensajes sin respuesta, se han habilitado los correspondientes temporizadores e interrupciones.

Para hacer un proceso de traspaso más completo, hemos permitido dos tipos de traspaso: uno manual y otro automático. En el primer caso, como su nombre indica, una vez los gestores han decidido traspasarse un cliente, solamente se le notifica al usuario de la estación cliente escogida que inicie la reasociación manualmente al AP indicado. En este caso, se trata de una sugerencia que se le hace al usuario y, puede éste, decidir si lo realiza o no. En el segundo caso es totalmente transparente y automático, apareciendo asociado al AP receptor correspondiente, si todo ha ido bien. En la figura 9 se muestra la información que se le muestra al usuario al producirse un traspaso correctamente con el mensaje enviado por el AP receptor.

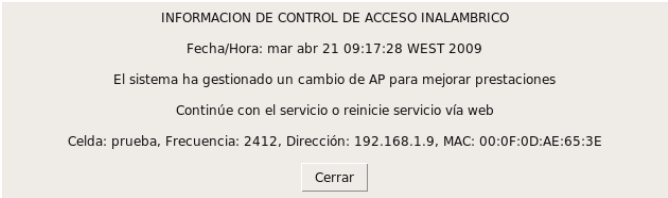


Fig. 9. Mensaje mostrado tras un traspaso automático

Destacar que en las múltiples pruebas realizadas para los traspasos de clientes se comprobó que los tiempos de transmisión (por ejemplo para las aplicaciones ftp, iperf, etc.) eran inferiores cuando se disponía de un uso exclusivo de celdas/canales o un menor número de clientes en el canal, tal y como pretendíamos con la funcionalidad de este módulo. Como ya se ha comentado, esta característica es especialmente necesaria cuando los servicios utilizados son dependientes del tiempo. Mantener los servicios activos durante el traspaso no son motivo de estudio en este trabajo, y en cualquier caso, deberían ser gestionados por las especificaciones del hand-off implementadas en el MAC 802.11.

V. CONCLUSIONES

En las redes WLAN que siguen el estándar IEEE 802.11 no se puede garantizar la calidad de servicio para determinadas aplicaciones, debido al número de estaciones participantes, las interferencias con otros equipos eléctricos/electrónicos,

infraestructuras/obstáculos existentes, la limitada velocidad, etc.

En este artículo se ha presentado una plataforma para reducir el tráfico inyectado desde la fuente si éste es menos prioritario que otros existentes en el mismo canal radio, y el traspaso dinámico de estaciones entre canales en función de la disponibilidad de canales y AP. En el primer caso se regula y distribuye el ancho de banda disponible y en el segundo se reubican las estaciones clientes allí donde se pueda obtener mejores resultados. En ambos casos, se realiza de forma transparente a los usuarios.

Con las pruebas realizadas hemos constatado que se ha logrado una mejora en la QoS ofrecida. Además, el tráfico de control usado para gestionar todo el sistema no ha afectado a las prestaciones de la red. Todo ello nos permite concluir que la inclusión de una plataforma basada en un control de varias funciones desde los AP con el gestor y aplicadas en las estaciones con los agentes, se logran mejoras en las prestaciones de las aplicaciones utilizadas y en la percepción del usuario. Además, la solución descrita es válida para cualquier producto o solución estándar sin cambiar su funcionamiento. De hecho, nuestra propuesta no pretende ser una solución que mejore las prestaciones de otras que modifiquen el control de acceso al medio sino que puedan ser complementarias a éstas, y se apliquen de forma combinada, pudiéndose integrar con otras más especializadas y en sistemas que incorporen otras mejoras internas a nivel de enlace, red o transporte.

Por último indicar que la funcionalidad del agente esta siendo mejorada para incluir más módulos que coexistan con los dos existentes. En esta línea se está estudiando un algoritmo de distribución de ancho de banda más elaborado, la gestión de los traspasos basados en localización, acciones dirigidas a la detección y reducción de las desconexiones, el uso del agente como intermediario en una red en malla, etc.

REFERENCIAS

[1] IEEE 802.11g-2003 IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications—Amendment 4: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band.

[2] IEEE 802.11b-1999 Supplement to IEEE 802.11-1999,Wireless LAN MAC and PHY specifications: Higher speed Physical Layer (PHY) extension in the 2.4 GHz band.

[3] IEEE 802.11s is a draft 802.11 amendment for mesh networking. www.ieee.org.

[4] IEEE 802.11e-2005, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements.

[5] PIEEE 802.11N (D2) Draft STANDARD for Information Technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications:Amendment : Enhancements for Higher Throughput.

[6] Yang Xiao, "IEEE:802.11e: QoS Provisioning at the MAC layer" IEEE Wireless Communications, Junio 2004.

[7] G. Boggia, P. Camarda, L. A. Grieco, S. Mascolo, "Feedback-based bandwidth allocation with call admision control for providing delay guarantees in IEEE802.11e networks", Noviembre 2004.

[8] F. Prihandoko, M. H. Habaebi, B. M. Ali, "Adaptive call admission control for QoS provisioning in multimedia wireless networks", Computer Communications, vol. 26, pp. 1560-1569, 2003.

[9] IntServ. RSVP, RFC2205. ftp://ftp.isi.edu/in-notes/rfc2205.txt.

[10] DiffServ RFC2475, http://www.ietf.org/rfc/rfc2475.txt.

[11] J. Jun, P. Peddabachagari, M. Sichitiu, "Theoretical Maximum Throughput of IEEE 802.11 and its Applications", *IEEE International Symposium on Network Computing and Applications (NCA)*, 2003, pp. 249-256.

[12] D. Marrero, A. Suárez, E. M. Macías, "Dynamic Interconnection of Ad-hoc Nodes Based on the Type of Service to be Accessed", *International Conference on Wireless Networks (ICWN)*, Junio 2005, pp. 539-545.

[13] D. Marrero, E. M. Macías, A. Suárez, "Dynamic Traffic Regulation for Wifi Networks". World Congress on Engineering 2007 (WCE2007). ICWN'07 Londres 2-4 Julio 2007, ISBN978-988-98671-2-6.

[14] Programming with pcap, http://www.tcpdump.org/pcap.htm.

[15] Netfilter/Iptables Project Homepage, http://www.netfilter.org/.

[16] D-Link Homepage, http://www.dlink.com/.

[17] Compaq Homepage http://www.compaq.com.

[18] Linux Advanced Routing & Traffic Control, www.lartc.org, http://tcng.sourceforge.net/.

[19] NLANR/DAST: Iperf 1.7.0 - The TCP/UDP Bandwidth Measurement Tool, http://dast.nlanr.net/Projects/Iperf/.

[20] VideoLAN - Free Software and Open Source Video Streaming Solution for Every OS., www.videolan.org.

Comportamiento de los usuarios WLAN en el campus UPC de Barcelona

Enrica Zola, Francisco Barcelo-Arroyo, María López-Ramírez
Departamento ENTEL,
Universitat Politècnica de Catalunya
C. Jordi Girona 1-3, 08034 Barcelona
enrica@entel.upc.edu, barcelo@entel.upc.edu, maria.lopez@entel.upc.edu

***Resumen*- A través del proyecto Education Roaming (eduroam), muchas universidades europeas permiten que sus usuarios puedan desplazarse entre ellas, disponiendo en todo momento de los servicios móviles igual que estuviesen en su propia universidad. La Universitat Politècnica de Catalunya (UPC) participa en este proyecto. Analizando los syslog de los puntos de acceso de la biblioteca principal del Campus Nord, se ha estudiado la actividad de los usuarios en la red WLAN de la UPC durante una semana y se ha extraído información sobre el comportamiento de los usuarios. A pesar de la difusión de dispositivos portátiles ligeros que facilitan su uso entre la gente joven mientras se va desplazando, en general los usuarios no se mueven mucho. No obstante la buena cobertura proporcionada por la infraestructura de red, los usuarios sufren muchos problemas de conectividad. Los resultados de nuestro trabajo pueden resultar útiles para mejorar la calidad de la red inalámbrica y para diseñar nuevas aplicaciones que se adapten a los hábitos de los usuarios.**

***Palabras Clave*- IEEE802.11, handover, medidas, tiempo de permanencia, WLAN.**

I. INTRODUCCIÓN

Desde finales de la década pasada, las comunicaciones inalámbricas se han difundido en muchos campos de la vida cotidiana y se han convertido en una tecnología de la que se dispone comúnmente en muchos entornos, tanto que la gente está acostumbrada a poderse mover mientras continúa comunicándose. A partir del año 2000, los usuarios se han mostrado cada vez más atraídos por la flexibilidad de esta tecnología, así que se ha registrado un incremento en el despliegue de redes inalámbricas en diferentes entornos. Las universidades han sido pioneras en ofrecer servicios WLAN en sus campus; un ejemplo es la Wireless Andrew en el campus universitario Carnagie Mellon [1], una red WLAN de banda ancha desarrollada en 1993. Actualmente, casi todas las universidades ofrecen conexión inalámbrica a sus estudiantes y trabajadores. Para estimular la movilidad de los investigadores y estudiantes europeos, en el 2003 se ha impulsado el proyecto eduroam (Education Roaming) [2] a través del cual se pretende ofrecer conexión inalámbrica en diferentes instituciones europeas; de esta manera, los trabajadores de empresas que participan en el proyecto pueden acceder a Internet a través de la WLAN de otra institución como si estuvieran en su propia empresa.

La popularidad creciente de las redes inalámbricas incentiva los investigadores a estudiar estos nuevos escenarios. Muchos trabajos han profundizado en el comportamiento de los usuarios y, especialmente, en las características del tráfico en entornos WLAN reales, como por ejemplo en los campus universitarios [4, 5, 6, 7, 8], en las empresas [9], durante una conferencia [10], o en entornos industriales [11]. En uno de las primeras publicaciones sobre este tema [6], Tang y Baker monitorizaron durante 12 semanas la red del departamento de Computer Science de la universidad de Stanford. Pese a que el estudio se base principalmente en el análisis del tráfico de la red, los autores presentan también unos primeros resultados sobre traspasos entre celdas (*handover*, HO): el número máximo de traspasos observados en un periodo de 5 minutos en un punto de acceso (AP) varía entre 2 y 10, según el AP; el AP de la biblioteca, por ejemplo, registra un máximo de 5 HO en un período de 5 minutos, y de 8 en 15 minutos.

Henderson, Kotz y Abyzov realizaron un análisis exhaustivo durante 17 semanas del cuadrimestre de otoño de 2003 en el Dartmouth College [5]. Entre los resultados de movilidad descubrieron que, con respecto al mismo análisis realizado en 2001 [4], los usuarios siguen sin moverse demasiado y tienden a quedarse en su *home location* por la mayoría del tiempo. Por *home location* ellos entienden el AP al que un usuario está asociado durante más de la mitad del tiempo total de permanencia en la red.

Un estudio parecido [9] se realizó en la WLAN de una empresa compuesta por tres edificios, monitorizando el tráfico entre el 20 de julio y el 17 de agosto de 2002. También aquí resultó que un gran porcentaje de usuarios no se movían mucho de su *home location*. Aquí el *home location* es el edificio con el AP más frecuentado, así que la movilidad se entiende entre edificios y no entre AP. Además, ya que usan el protocolo SNMP para obtener los datos, los resultados de movilidad pueden estar afectados por el tiempo de interrogación (*poll time* de 5 minutos). En nuestro trabajo no hemos utilizado el protocolo SNMP así que no tenemos el inconveniente de perder los movimientos entre períodos de *poll*.

Un estudio interesante sobre el traspaso entre celdas en un entorno real [7] se basa en los datos recogidos en 1997 de la Wireless Andrew del Carnagie Mellon. El estudio está basado en 9105 muestras de tiempos de permanencia, de los cuales el 54% es inferior a 3 segundos y el 93% es inferior a 5 minutos. Sin embargo, el valor medio es de 50 minutos. En nuestro estudio se han encontrado resultados parecidos, pero se intentará tomar un punto de vista distinto para dar una interpretación del porqué el valor medio y la mediana del tiempo de permanencia son tan diferentes. Además, en el análisis del tiempo entre nuevas llegadas [7] (es decir,

nuevos usuarios) no se entiende bien si los autores han tenido en consideración los ciclos diarios.

En este artículo se presentan los resultados obtenidos a partir de la información de asociación recogida durante una semana en la biblioteca del campus Nord de la UPC en Barcelona. A diferencia de otros trabajos previos, nos concentramos en los resultados de movilidad de los usuarios de la biblioteca. El objetivo es caracterizar mejor el tiempo de permanencia en una celda (el tiempo entre HO) e identificar comportamientos diferentes entre usuarios. Ya que se usa la herramienta syslog para obtener la información desde los AP, no perdemos información como en otros trabajos [9].

El resto del documento se organiza de la siguiente manera. En la Sección II se describe cómo se han recogido los datos. La Sección III da mayores detalles sobre el comportamiento de los usuarios. Los resultados del tiempo de permanencia en una celda se recogen en la Sección IV. La Sección V concluye el trabajo.

II. RECOGIDA DE DATOS

El estudio se ha realizado en la Universitat Politècnica de Catalunya (UPC): ésta se compone de numerosos campus en Barcelona y en sus alrededores. Para facilitar la movilidad de investigadores y estudiantes europeos, la UPC participa en el proyecto eduroam [2] que proporciona conectividad inalámbrica tanto a sus usuarios como a los de otras instituciones que participan en el proyecto. Con esta iniciativa, algunas organizaciones permiten que sus usuarios puedan desplazarse entre ellas, disponiendo en todo momento de los servicios móviles que pudieran necesitar. El objetivo es garantizar a los usuarios que lleguen a otra organización de disponer, de la manera más transparente posible, de un entorno de trabajo virtual con conexión a Internet, acceso a servicios y recursos de su organización origen, así como acceso a servicios y recursos de la organización que en ese momento les acoge. Los únicos requerimientos son usar un dispositivo con acceso Wi-Fi compatible con la tecnología IEEE 802.11b o 802.11g, y disponer de las credenciales de acceso otorgadas por la universidad de origen.

En este trabajo se ha analizado el tráfico WLAN IEEE802.11 de la biblioteca ubicada en el Campus Nord de Barcelona. El edificio de la biblioteca tiene 4 plantas y, en cada planta, hay dos puntos de acceso (AP). Esta infraestructura proporciona una buena cobertura a todo el edificio: como ejemplo, véase la Figura 1 en la que se muestra el mapa de cobertura del primer piso de la biblioteca y la ubicación de los dos AP (AP 101 y AP 102). La entrada a la biblioteca está ubicada en la planta baja, donde hay un servicio de préstamo de libros y ordenadores portátiles. El primer y segundo piso albergan la colección de la biblioteca, repartida por temas; los estudiantes pueden quedarse en estos dos pisos para estudiar, consultar los libros y navegar por Internet. En el tercer piso se puede encontrar bibliografía más específica para estudiantes de doctorado e investigadores.

Los usuarios de la biblioteca no han sido avisados del estudio que se iba a realizar. La única información sensible con la que hemos tratado fueron la dirección MAC y la dirección IP de los equipos, así como los nombres asignados

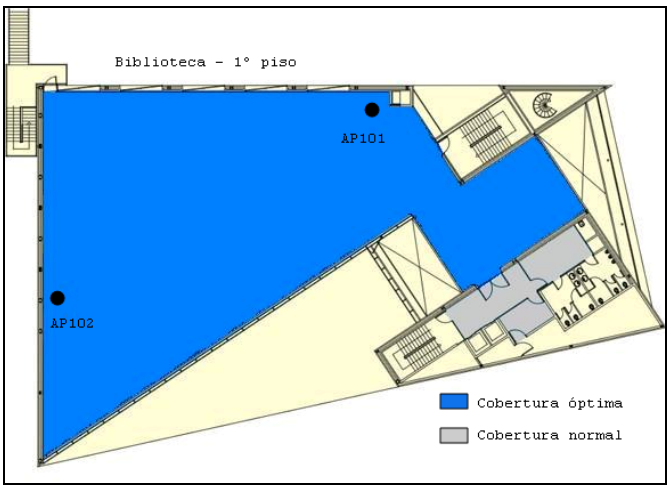


Fig. 1. Plano de cobertura del primer piso de la biblioteca y posición de los dos AP.

a los AP. Para mantener la privacidad, las trazas fueron anonimizadas antes de trabajar con ellas.

A. Syslog

La información de asociación necesaria para este análisis se ha obtenido con syslog, un estándar que permite reenviar los mensajes en una red IP. Syslog es un protocolo cliente/servidor en donde el cliente envía pequeños mensajes de texto al servidor a través de UDP o de una conexión TCP. Se han configurado los AP de la WLAN de la UPC para que envíen mensajes syslog a un servidor central cada vez que un dispositivo WLAN se autentica o deautentica, se asocia o desasocia, o hace un traspaso. Cada uno de estos mensajes contiene el nombre del AP, la dirección MAC del dispositivo, el instante en el que el AP ha recibido el mensaje (con una precisión de un segundo), y el tipo de mensaje. Una vez que se han anonimizado los mensajes, se pueden procesar los mensajes de asociación y desasociación.

Un usuario WLAN que quiera conectarse a la infraestructura, escanea el medio en búsqueda de un AP al que asociarse y escoge el mejor en términos de potencia de señal recibida: antes de asociarse, el usuario debe autenticarse con dicho AP. Una vez asociado, puede intercambiar información dentro de la red a través del AP hasta que se desasocie de él. La desasociación puede ser debida a un traspaso a otra celda cubierta por otro AP, a problemas de autenticación (en este caso, el mensaje de desasociación tiene como motivo *Previous Authentication no longer valid*) o porque el usuario ha abandonado la red.

B. Configuración

El objetivo de este estudio es analizar el tiempo de permanencia en una celda en un entorno real: para eso, se han considerado las tramas de asociación y desasociación. Con la fórmula siguiente se puede calcular el tiempo de permanencia en una celda para cada usuario (es decir, el tiempo que un usuario está asociado a un AP):

$$cell \quad residence \quad time = TS_{des} - TS_{as}, \qquad (1)$$

donde TS_{as} es el *timestamp* de la asociación de un usuario a un determinado AP (es decir, el instante en el que el AP recibe la petición de asociación del usuario) y TS_{dis} es el *timestamp* de la desasociación del mismo usuario de aquel

AP (es decir, el instante en el que el AP recibe la trama de desasociación enviada por el usuario). Si un AP recibe dos o más peticiones de asociación del mismo usuario, se ha tenido en cuenta la última ya que puede darse que el mensaje de confirmación de asociación enviado por el AP se haya perdido y el usuario haya enviado otra vez su petición de asociación a dicho AP.

Este estudio se basa en la información de asociación recibida por los AP de la biblioteca durante una semana, desde el lunes 2 de junio hasta el viernes 6 de junio de 2008. Esta semana precedió los exámenes finales, con lo que la actividad registrada es suficientemente alta como para obtener resultados estadísticos consistentes.

III. COMPORTAMIENTO DE USUARIOS

Un total de 1085 direcciones MAC diferentes se han asociado a algún AP de la biblioteca durante el periodo analizado. El número de usuarios podría ser incluso mayor, ya que la biblioteca ofrece en préstamo ordenadores portátiles a los estudiantes, con lo que usuarios diferentes pueden haber usado el mismo portátil (es decir, la misma dirección MAC). Por simplicidad, en el resto del documento nos referimos a usuario en lugar de hablar de dirección MAC.

La Fig. 2 representa la distribución por hora del número de primeras asociaciones de cada día a cualquier AP de la biblioteca (es decir, el primer instante de cada día en el que un nuevo usuario entra en la biblioteca y se conecta a la WLAN). La biblioteca está abierta de 8.30 de la mañana hasta las 2.30 de la noche; efectivamente no hay asociaciones fuera de este período. Cuando la biblioteca abre, hay un primer pico en el histograma; durante el resto del día, nuevos usuarios continuamente acceden a la biblioteca, aunque el número vaya disminuyendo. Hay nuevos usuarios también después de las 8 de la noche y hasta la medianoche, así que podemos decir que hay actividad en la biblioteca también fuera del horario laboral: sin embargo el patrón nocturno varía durante la semana, así que se ha preferido limitar el análisis al patrón de jornada laboral (es decir, de 10 a.m. hasta 5 p.m.) para trabajar con datos homogéneos. Lunes es el día con más usuarios nuevos, mientras que el miércoles es el que menos.

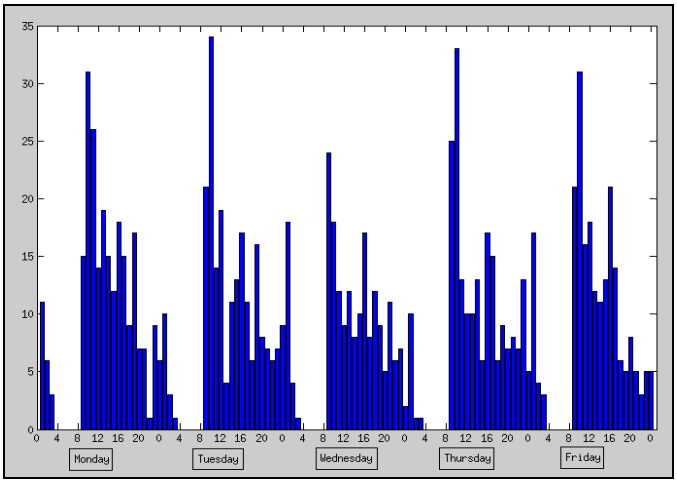


Fig. 2. Número de primeras asociaciones por día y hora.

En la Tabla 1 se muestra el número de nuevos usuarios que se asocian a algún AP entre lunes y viernes: el mismo usuario se puede asociar más de una vez a un AP durante el día, pero aquí se tiene en cuenta sólo la primera vez (es decir, la primera asociación que se registra a aquel AP por día). En la segunda columna se muestra el valor sobre el período entero (de lunes a viernes, de 0 a 24h), mientras que la tercera columna contempla el patrón de día laboral (de lunes a viernes, de 8 a 17h). La última línea de la tabla indica el número de nuevos usuarios que se asocian a cualquier AP de la biblioteca: aquí un usuario que se mueva y se vaya asociando a más de un AP sólo se cuenta una vez, por lo tanto no es la suma de las otras celdas de la tabla. Se puede observar que la proporción de usuarios que aparecen durante el patrón de día laboral con respecto a los que aparecen durante todo el período es siempre alrededor del 50%, excepto por los AP del segundo piso (AP 201 y AP 202) que registran un 33%. El 60% de las primeras asociaciones se registran en los AP del primero y segundo piso (AP 101, 102, 201, 202): esto puede ser debido a que en estas plantas los estudiantes disponen de mesas y del acceso a Internet. Los AP 102 y 202 han sido escogidos para el análisis del tiempo de permanencia ya que son los puntos de acceso más cargados durante la jornada laboral.

Para caracterizar ulteriormente la población de la biblioteca, se ha analizado la frecuencia con la que los usuarios acceden a la WLAN: en la Tabla 2 se presentan los porcentajes de usuarios que acceden uno o más días. Más de la mitad de los usuarios acceden sólo un día en toda la semana, así que nuestra población está compuesta por una gran cantidad de usuarios no-frecuentes. Este porcentaje disminuye a medida que aumentamos el número de días: sólo un 5,34% accede cada día. Si además se tiene en cuenta que la biblioteca presta ordenadores portátiles, posiblemente

Número de usuarios entre: / AP:	0-24h	10-17h
001	316	160
002	89	57
101	300	151
102	431	230
201	411	136
202	476	164
301	211	108
302	187	111
Toda la biblioteca	1085	473

Tabla 1. Número de usuarios por AP de lunes a viernes.

Número de días	Número de usuarios	%
1	355	57,44
2	136	22,01
3	62	10,03
4	32	5,18
5	33	5,34

Tabla 2. Porcentaje de usuarios que acceden uno o más días en la biblioteca.

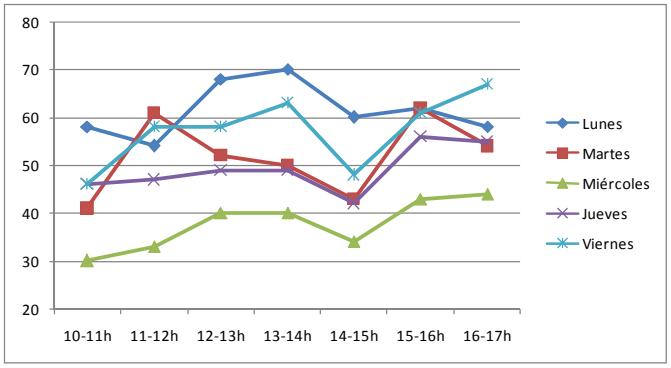


Fig. 3. Número de usuarios conectados en horario laboral a cualquier AP de la biblioteca por día y hora.

estemos contando como usuarios frecuentes los usuarios de algunos portátiles de la biblioteca. Ya que un usuario que aparece más de un día es contado sólo una vez, la suma de los valores de la Tabla 2 no coincide con el número de usuarios totales (es decir, 1085) en la biblioteca presentado en la Tabla 1.

El número de usuarios conectados a cualquier AP de la biblioteca por día y hora está representado en la Fig. 3. Lunes es el día con más usuarios, mientras que el miércoles el que menos, así que las figuras mostradas en la Fig. 2 para las primeras asociaciones se mantienen con el total de asociaciones. Cada día, excepto el martes, el número de usuarios conectados crece entre 11 y 13 horas y disminuye a la hora de comer (entre 14 y 15 horas). Todos los días menos el lunes, el número de usuarios asociados crece por la tarde. El número elevado de usuarios poco frecuentes no influye en el patrón de jornada laboral de las nuevas llegadas presentadas en la Fig. 2, tal como se observa también en [9]. Por otro lado, el comportamiento de los usuarios en términos de permanencia en la biblioteca cambia durante la semana y tiene un impacto sobre los resultados de la Fig. 3.

IV. Tiempo de permanencia en una celda

Para usar datos homogéneos, se han analizado las trazas de cada AP durante la jornada laboral (de lunes a viernes entre 10 a.m. y 5 p.m.). Las tramas MAC IEEE802.11 de asociación y desasociación de cada AP han sido analizadas y los AP 102 y 202 han sido elegidos como más representativos (alta y constante carga durante la jornada laboral). A partir de cada conjunto de datos, hemos obtenido un total de 1036 y 929 tiempos de permanencia, respectivamente.

La Tabla 3 muestra las estadísticas del tiempo de permanencia total en una celda: el valor medio para el AP 102 es de 416 segundos (cerca de 7 minutos) y la desviación estándar es de 1246 segundos (cerca de 21 minutos), con un coeficiente de variación elevado (CV próximo a 3). Para el AP 202 la media es más baja (295 segundos, cerca de 5 minutos) y el CV es superior (3,38). A pesar de que el valor medio sea alto, el percentil 50 es muy bajo (alrededor de 1 minuto por cada AP), reflejando un alto porcentaje de tiempos de permanencia en la celda muy cortos.

Debido al solapamiento en cobertura entre celdas vecinas, es frecuente que un usuario se encuentre en la situación representada en la Fig. 4: un dispositivo que se encuentre en el primer piso está asociado, por ejemplo, al

	AP102			AP202		
	Total	**No HO**	**HO**	**Total**	**No HO**	**HO**
Media	416	1480	273	295	631	259
Mediana	86	218	76	66	136	61
Máximo	14339	14339	10389	15558	15558	11976
Desv. estándar	1246	2668	798	995	1736	875
CV	2,99	1,80	2,93	3,38	2,75	3,38
Número de muestras	1036	123 11,87%	913	929	89 9,58%	840

Tabla 3. Estadísticas del tiempo de permanencia para el AP 102 y el AP 202.

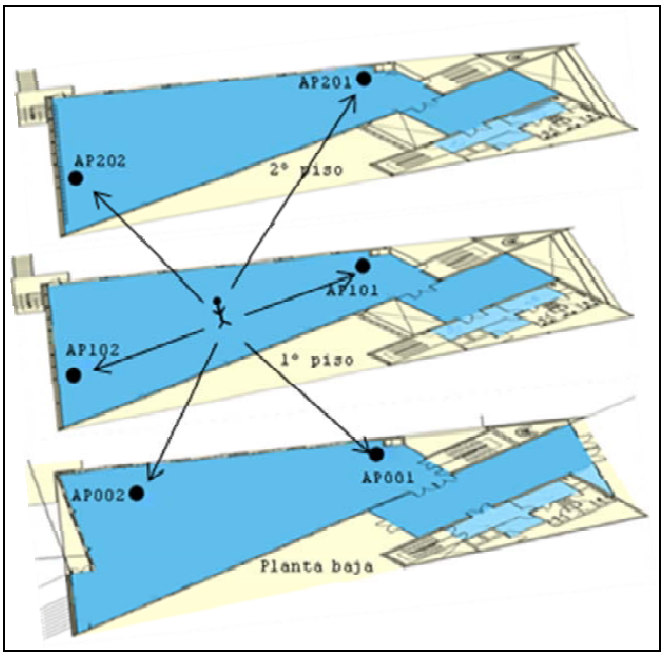


Fig. 4. Ejemplo de usuario bajo el efecto ping-pong.

AP 102; sin embargo, el nivel de la señal recibida por los otros AP (AP 101, 201, 202, 001 y 002) puede ser mejor que el del AP102. El usuario podría sufrir el efecto ping-pong entre AP cercanos, a pesar de no estar moviéndose. Ésta es una situación muy común en una red inalámbrica con una alta concentración de AP (es decir, con mucha sobrecobertura). En la biblioteca muchos usuarios parecen estar sufriendo este fenómeno y, por eso, continuadamente se asocian y desasocian con AP diferentes: está claro que la desasociación de un AP para asociarse con otro AP vecino no siempre implica un verdadero movimiento del usuario, como ya se observó en [5].

Debido a la imposibilidad de discernir entre movimientos reales y efecto ping-pong ya que los logs no proporcionan información de la posición de los usuarios, se ha hecho un seguimiento de los usuarios que tan sólo se asocian a un AP en todo el día (es decir, los usuarios que no hacen ningún traspaso entre celdas) y se ha estudiado el tiempo de permanencia en la celda de estos usuarios (No HO) por separado del resto de usuarios. Los resultados se presentan en la Tabla 3 (No HO): como esperable, se han obtenido

valores medios más altos (por ej. alrededor de 24 y 10 minutos para el AP 102 y 202, respectivamente) y resultados más estables para ambos AP (los valores CV son inferiores). El percentil 50 es todavía muy bajo (3,5 y 2 minutos, respectivamente), reflejando que todavía hay muchos tiempos de permanencia muy cortos a pesar de que estos usuarios no sufran el efecto ping-pong entre celdas vecinas. La Tabla 3 muestra también los resultados para los usuarios que sí hacen algún traspaso a otra celda (HO): el valor medio es alrededor de 4,5 minutos para ambos AP y los CV siguen siendo elevados. Es interesante observar que los usuarios HO representan el 90% de la muestra total: esto implica que en la red hay mucha señalización relacionada con las peticiones de traspaso de celda.

El número medio de HO por AP y hora es fácilmente deducible de la Tabla 3: en el caso de la muestra total, eso varía entre 9 y 12 según el AP, mientras que para el caso de usuarios HO es estable alrededor de 13.

Las Fig. 5 y 6 muestran la distribución del tiempo de permanencia en la celda para el AP 102 y para los dos casos de usuarios No Ho y HO, respectivamente: se ha usado la escala logarítmica para poder observar mejor la alta concentración de conexiones cortas (inferiores a 10 minutos) sin perder la cola (los valores máximos se pueden ver en la Tabla 3). Los resultados obtenidos son muy diferentes de los presentados en [7], donde el 54% de los tiempos de permanencia eran inferiores a 3 segundos; sin embargo, nuestros resultados demuestran que todavía hay problemas de conectividad que no son imputables a una mala planificación de cobertura o a la movilidad de los usuarios. Es necesario profundizar sobre las causas que hacen que un usuario no disponga de un servicio continuo a pesar de que no se mueva.

V. CONCLUSIONES

El comportamiento de los usuarios de la WLAN del Campus Nord de la UPC de Barcelona ha sido estudiado a través de los logs recogidos en los AP de la biblioteca durante la semana entre el 2 y el 6 de junio de 2008. Los usuarios se reparten bastante ecuamente en toda la biblioteca, pero el 60% de las primeras asociaciones se registran en los AP de la primera y segunda planta. Además, el 60% de los usuarios se conectan a la WLAN sólo un día en toda la semana; a pesar de tener usuarios poco frecuentes, sin embargo el patrón de jornada laboral se mantiene durante todo el período analizado.

A partir del tiempo de permanencia en una celda, se ha deducido que los usuarios no se mueven mucho: el 12% de toda la población sólo se conecta a un AP por día (usuarios No HO). No ha sido posible extraer información sobre la posición de los usuarios o sobre su movilidad, pero se ha estudiado el tiempo medio de conexión a un AP (tiempo de permanencia en una celda) y se ha visto que es muy variable. Si por un lado hay un alto porcentaje de conexiones inferiores a 1 minuto, por otro lado hay usuarios que permanecen asociados a un AP durante casi 4 horas. Se ha decidido analizar separadamente la población de usuarios No HO y la de usuarios HO para poder deducir las diferentes tendencias. El tiempo medio de permanencia en una celda es de 4 minutos para los usuarios HO y de entre 10 y 25 minutos, dependiendo del AP, para los usuarios No HO. Sin embargo, también para los usuarios No HO, que por definición no cambian de AP durante todo el tiempo que están en la biblioteca, hay muchos tiempos de permanencia muy cortos. Esto nos hace concluir que hay algún problema de conectividad que no está relacionado con la movilidad ni con una mala cobertura y nuestro objetivo futuro es investigar las causas de estos tiempos cortos de conexión a un AP.

AGRADECIMIENTOS

Nuestros agradecimientos al equipo de UPCnet que nos ha proporcionado las trazas de la WLAN de la UPC; los autores quieren agradecer en especial manera a Sergi Sales Llop quien se ha encargado de gestionar la colaboración, y a Josep-Lluís Cortés y Margarita Garrido Lorenzo por su ayuda y soporte técnico.

Este trabajo ha sido apoyado por el Ministerio de Ciencia y Tecnología del Gobierno de España y el FEDER a través del proyecto CICYT TEC2006-09466/TCM.

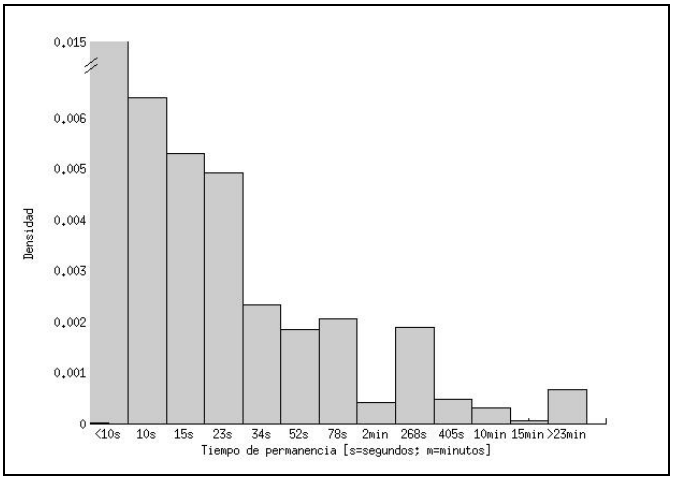


Fig. 5. Distribución del tiempo de permanencia en una celda para los usuarios No HO en el AP 102.

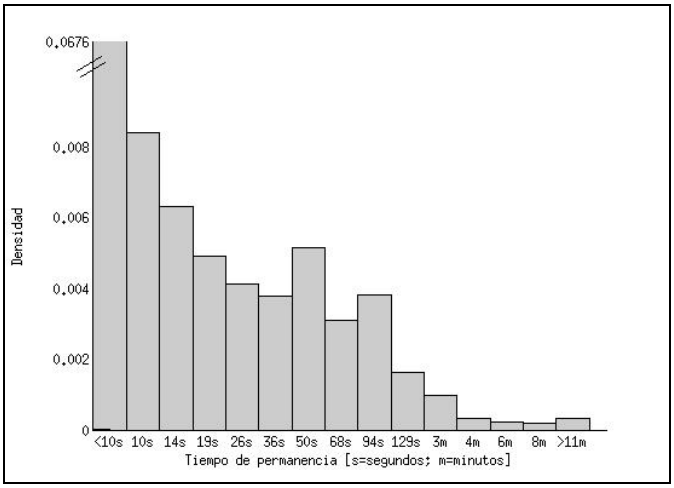


Fig. 6. Distribución del tiempo de permanencia en una celda para los usuarios HO en el AP 102.

REFERENCIAS

[1] A. Hills, "Wireless Andrew [Mobile Computing for University Campus]," *IEEE Spectrum*, vol. 6, pp. 49-53. DOI= 10.1109/6.769269 (1999)

[2] Eduroam project at UPC, https://upcnet.upc.edu/serveis/servidors-i-xarxes/gestio-de-xarxes/xarxes-sense-fils-upc-eduroam/xsf-upc-eduroam-upc-wireless

[3] K. Wierenga and L. Florio, "Eduroam: past, present and future," *Computational Methods in Science and Technology,* vol. 11 (2), pp. 169-173 (2005)

[4] D. Kotz, T. Henderson and I. Abyzov, "Analysis of a Campus-wide Wireless Network," *Wireless Networks*, vol. 11, pp. 115-133 (2005)

[5] T. Henderson, D. Kotz and I. Abyzov, "The Changing Usage of a Mature Campus-wide Wireless Network," *Computer Networks*, vol. 52, pp. 2690-2712 (2008)

[6] D. Tang and M. Baker, "Analysis of a Local-Area Wireless Network," Proc. 6th Annual International Conference on Mobile Computing and Networking, pp. 1–10, ACM Press, Boston (2000)

[7] S. Thajchayapong and J.M. Peha, "Mobility Patterns in Microcellular Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 5 no. 1, pp. 52-63 (2006)

[8] R. Hutchins and E.W. Zegura, "Measurements from a Campus Wireless Network," Proc. IEEE International Conference on Communications, ICC 2002, vol. 5, pp. 3161-3167 (2002)

[9] M. Balazinska and P. Castro, "Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network," Proc. 1st International Conference on Mobile Systems, Applications and Services, pp. 303-316 (2003)

[10] A. Balachandran, G.M. Voelker, P. Bahl and P.V. Rangan, "Characterizing User Behavior and Network Performance in a Public Wireless LAN," Proc. 2002 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, pp. 195–205, ACM Press (2002)

[11] A. Willig, M. Kubisch, C. Hoene and A. Wolisz, "Measurements of a Wireless Link in an Industrial Environment Using an IEEE 802.11-Compliant Physical Layer," *IEEE Transactions of Industrial Electronics*, vol. 49, no. 6, pp. 1265-1282 (2004)

Adaptabilidad de enlace en sistemas IEEE 802.11n

Gabriel Martorell, Felip Riera-Palou y Guillem Femenias
Grupo de Comunicaciones Móviles,
Universitat de les Illes Balears
Ctra. Valldemosa Km 7.5
07122 Palma de Mallorca
gabriel.martorell, felip.riera, guillem.femenias@uib.es

***Resumen*—En este artículo se estudia el uso de algoritmos de adaptación rápida del enlace (FLA) en el contexto de un sistema de múltiples antenas en transmisión y recepción (MIMO) como es el *draft* del estándar de la próxima generación de redes locales inalámbricas (WLANs) IEEE 802.11n. FLA selecciona el esquema de codificación y modulación (MCS) más apropiado a las características instantáneas del canal para satisfacer unos determinados requerimientos de servicio. La técnica FLA presentada se basa en relacionar la relación señal a ruido (SNR) efectiva exponencial (EESM) a tasa de error por paquete (PER). Además, se propone un esquema FLA basado en la tasa de error de bit (BER) que simplifica el procedimiento de calibración sin provocar una degradación de prestaciones. Los resultados demuestran que ambas técnicas FLA (las basadas en PER y BER), seleccionan de manera casi óptima el MCS que maximiza la capacidad, mientras satisfacen una PER objetivo preestablecida. Además, se han considerado los efectos de los errores en la estimación de canal, quedando patente la importancia de disponer de buenos estimadores para que los algoritmos FLA puedan funcionar correctamente.**

***Index Terms*—FLA, Algoritmos de adaptación rápida del enlace, MIMO-OFDM, IEEE 802.11n, EESM, cross-layer.**

I. INTRODUCCIÓN

Durante la última década el concepto de computación móvil se ha hecho una realidad debido al espectacular auge experimentado en el despliegue de redes inalámbricas de área local (WLANs), que a su vez se debe en gran medida al éxito de los estándares IEEE 802.11a/g. Como todos los sistemas inalámbricos, las WLANs deben combatir efectos del canal radio tales como la selectividad frecuencial y/o temporal. Para contrarrestar dichos efectos la capa física de IEEE 802.11a/g utiliza: 1) multiplexación por división en frecuencias ortogonales (OFDM), y 2) codificación y modulación adaptativa (AMC) basada en BICM (*Bit Interleaved Coded Modulation*).

Idealmente, las estrategias AMC que utilizan información sobre el estado del canal (CSI) intentan seleccionar una combinación de esquemas de modulación y codificación (MCS) con el objetivo de optimizar la eficiencia espectral, sujeta a los requerimientos de calidad de servicio (QoS) tales como, por ejemplo, una probabilidad media máxima de *outage* relativa a la tasa de error por paquete (PER), P_{out}, para una PER objetivo dada, PER_0. Matemáticamente, la PER instantánea del MCS $m \in \boldsymbol{\mathcal{M}}$, donde $\boldsymbol{\mathcal{M}}$ es el conjunto de MCSs, se expresa como

$$P_m = \mathcal{F}_m\left(\text{SNR}, L, \boldsymbol{H}\right), \tag{1}$$

donde SNR es la relación señal a ruido en recepción, L es la longitud del paquete en número de bits de información y $\boldsymbol{H}$ denota la realización de canal. El proceso de selección óptima del MCS puede entonces formularse como

$$\begin{aligned} m^* &= \arg \max_{m \in \boldsymbol{\mathcal{M}}} \eta_m = \arg \max_{m \in \boldsymbol{\mathcal{M}}} T_m \left(1 - P_m\right) \\ &\text{sujeto a} \\ & Pr\left\{\text{PER} > PER_0\right\} \leq P_{\text{out}}, \end{aligned} \tag{2}$$

donde η_m y T_m indican respectivamente el *throughput* instantáneo y la tasa de transmisión del MCS m. Este proceso de optimización, conocido como adaptación rápida del enlace (FLA), conlleva una interacción entre la capa física (PHY) y la capa de control de acceso al medio (MAC), requiriendo de un diseño *cross-layer* (intercapas) PHY-MAC. Desafortunadamente, los estándares actuales WLAN (802.11 a/g) sólo especifican el conjunto de MCS permitidos en cada tipo de tramas MAC, pero no cómo y cuándo se debe cambiar de tasa. Además, no existe ningún mecanismo de señalización en recepción que posibilite el envío al transmisor de información sobre la calidad actual del enlace o la tasa a utilizar, imposibilitando la implantación de técnicas FLA. Como alternativa a FLA, Kamerman y Montean [1] describen el protocolo adaptativo de enlace AutoRate Fallback (ARF) aplicable a los estándares WLAN actuales. Mediante este algoritmo la capa física disminuye la tasa de transmisión de forma automática después de que se produzcan dos errores consecutivos (ACKs perdidos) y aumenta la tasa de transmisión después de diez transmisiones correctas (ACK recibidos) o después de un *time out*. Esta propuesta y sus variantes (ver, por ejemplo, [2], [3], [4], [5], [6]) explotan el hecho de que dada una SNR y un conjunto de MCS, la selección de tasas de transmisión más altas en el MCS implican una PER instantánea superior. La ventaja principal de estos algoritmos es su simplicidad de implementación. Las desventajas son su funcionamiento subóptimo y sus pobres prestaciones cuando se producen colisiones [7].

Actualmente, el IEEE 802.11 *High Throughput Task Group committee* está llevando a cabo la estandarización de la siguiente generación de WLANs, denominada IEEE 802.11n [8]. El nuevo estándar alcanzará tasas de transmisión más altas gracias a la utilización de múltiples antenas en transmisión y recepción (MIMO) y otras mejoras tales como la posibilidad de operar en una banda de 40 MHz (utilizando más subportadoras) y modos de transmisión con intervalos de guarda reducidos. Adicionalmente, su capa MAC incorpora mecanismos para enviar al transmisor información respecto del MCS seleccionado, convirtiendo FLA en una opción viable. Algunos ejemplos del uso de FLA en sistemas MIMO-OFDM son los presentados en [9] y [10]. En los sistemas MIMO, además de la selección del MCS, los algoritmos FLA

deben también seleccionar el modo MIMO: códigos bloque espacio-temporales (STBC), diversidad por retardo cíclico (CDD), el modo de multiplexación por división espacial (SDM) o una combinación de ellos. En este caso, y debido a la componente MIMO, tasas de transmisión superiores no implican necesariamente una PER instantánea superior y por lo tanto, los algoritmos tradicionales de adaptación del enlace utilizados en los sistemas *single-input multiple-output* (SIMO) no son efectivos. Por este motivo, es necesario el desarrollo de algoritmos de adaptación del enlace específicos para sistemas MIMO-OFDM. Como se aprecia en la ecuación (2), los elementos principales del proceso de optimización son, por una parte, una herramienta de predicción de PER en la capa física para todos los possibles modos MCS/MIMO, longitud de paquetes y realizaciones de canal, y por otro lado, una metodologia de selección del modo MCS/MIMO en la capa MAC que asegure el cumplimiento de los requisitos de QoS. No existe una aproximación simple y sistemática para predecir la PER asumiendo modos MCS/MIMO arbitrarios, cualquier tamaño de paquete y realizaciones de canales selectivos en frecuencia con correlaciones de canal arbitrarias. No obstante, este artículo propone técnicas de abstracción de la capa física que permiten la predicción de PER basada en la predicción trama a trama de la tasa de error por bit (BER). Las técnicas de abstracción están basadas en una aproximación que asigna parámetros de sistema, tales como el modo operacional MCS/MIMO seleccionado, la longitud de paquete y la realización de canal, a una métrica de calidad del enlace (LQM) que puede asociarse a la PER mediante simples tablas [10], dependiendo esta LQM de la estrategia de detección MIMO utilizada en el receptor. En este artículo se presenta una nueva metodología para la selección del MCS que satisface las restricciones de optimización sobre una probabilidad de servicio PER. Además, se puede intuir que la precisión de la estimación de CSI tiene un papel importante en las prestaciones de los algoritmos de adaptación del enlace, por esta razón, también presentamos un estudio del impacto sobre el *throughput* de las imperfecciones en la estimación de canal en redes IEEE 802.11n.

En la Sección II se describe el modelo de sistema utilizado en el artículo, correspondiente al IEEE 802.11n. Seguidamente, en la Sección III se explica el funcionamiento de los algoritmos FLA basados en la predicción de la PER, el algoritmo utilizado para determinar el MCS óptimo y un novedoso sistema FLA que se basa en la predicción de la BER. En la sección IV se presentan los resultados obtenidos con la aplicación de ambos sistemas FLA (PER y BER) para diferentes modelos del canal de propagación y tomando en consideración los posibles efectos de una estimación no ideal de la respuesta frecuencial del canal. Finalmente, la sección V resume las conclusiones del artículo.

II. Modelo de sistema

II-A. Transmisor

Nuestro estudio se centra en la propuesta de estandarización IEEE 802.11n [8] que tiene como modelo de transmisor el diagrama de bloques de la Fig. 1. Inicialmente, los bits de información $\{b_1, b_2, \ldots, b_L\}$ son codificados convolucionalmente con una tasa de codificación $R = \frac{1}{2}$ y seguidamente se perforan a una de las tasas de codificación definidas $R_m \in \{1/2, 2/3, 3/4, 5/6\}$. En función de la configuración MIMO seleccionada, los bits obtenidos son demultiplexados en N_s *streams* espaciales que seguidamente son procesados independientemente. Para cada *stream*, los bits codificados se entrelazan y se convierten a símbolos de una de las posibles constelaciones (BPSK, QPSK, 16-QAM o 64-QAM). A continuación, y dependiendo del modo MIMO activo, los símbolos se codifican usando STBC o directamente se asignan a una de las N_T antenas transmisoras (SDM). Sobre cada uno de los *streams* espaciales se puede aplicar un retardo cíclico que, si existen suficientes antenas en transmisión, puede incrementar la diversidad frecuencial del sistema. Finalmente, los símbolos resultantes se envían a un modulador OFDM convencional que consiste en una IFFT y la adición de un intervalo de guarda completo.

Para simplificar la presentación, este artículo se centra en una configuración del IEEE 802.11n con un transmisor de N_T=2 antenas y un receptor de $N_R = 2$ antenas, lo que sugiere que los MCS con *streams* espaciales $N_s = 1$ y $N_s = 2$ utilicen STBC [11] y SDM [12], respectivamente. No se aplica CDD.

II-B. Receptor

La Fig. 2 muestra el diagrama de bloques para un posible receptor genérico de IEEE 802.11n. El proceso de recepción empieza invirtiendo el proceso de modulación OFDM (p.e. eliminación del GI y procesamiento FFT) para recuperar las muestras recibidas en banda base. Las muestras sobre la subportadora k-ésima en el instante t se pueden expresar como:

- SDM

$$\boldsymbol{r}_t[k] = \boldsymbol{H}_t[k]\boldsymbol{s}_t[k] + \boldsymbol{\eta}_t[k] \tag{3}$$

- STBC

$$\boldsymbol{r}_t[k] = \boldsymbol{H}_t[k] \begin{bmatrix} s_{t,1}[k] \\ s_{t,2}[k] \end{bmatrix} + \boldsymbol{\eta}_t[k] \tag{4}$$

$$\boldsymbol{r}_{t+1}[k] = \boldsymbol{H}_{t+1}[k] \begin{bmatrix} -s^*_{t,2}[k] \\ s^*_{t,1}[k] \end{bmatrix} + \boldsymbol{\eta}_{t+1}[k] \tag{5}$$

donde $\boldsymbol{H}_t[k]$ denota la matriz de canal MIMO $N_R \times N_T$ para la subportadora k, $\boldsymbol{s}_t[k] = [s_{t,1}[k], s_{t,2}[k]]^T$ es el vector de símbolos transmitidos con $E\{\boldsymbol{s}_t[k]\boldsymbol{s}_t[k]^H\} = (P_T/N_T)\boldsymbol{I}_{N_T}$, $\boldsymbol{I}_{N_T}$ es la matriz identidad de dimensión $N_T \times N_T$ y $\boldsymbol{\eta}_t[k]$ es el vector de ruido térmico $N_R \times 1$ caracterizado como un ruido Gaussiano blanco aditivo de media cero y matriz de covarianza $E\{\boldsymbol{\eta}_t[k]\boldsymbol{\eta}_t[k]^H\} = \sigma^2\boldsymbol{I}_{N_R}$. Es importante remarcar que estas definiciones permiten expresar la relación señal-ruido para cada antena receptora como $E_s/N_0 = \frac{P_T}{N_T\sigma^2}$.

Si $N_s = 1$, se aplica STBC en el transmisor. Esta técnica realiza una combinación ortogonal de dos símbolos OFDM consecutivos (bloque STBC) que permiten detección óptima en el receptor por medio de *maximal ratio combining* (MRC), el cual es implementado por una combinación lineal de muestras recibidas en dos símbolos OFDM consecutivos. De este modo, asumiendo CSI ideal y que el tiempo de coherencia es suficientemente largo como para asegurar que

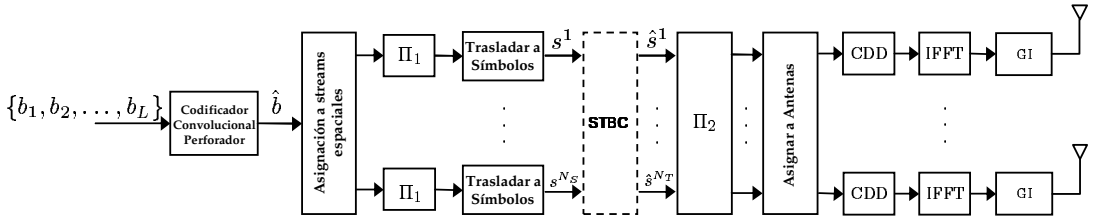


Figura 1. Diagrama del transmisor IEEE 802.11n

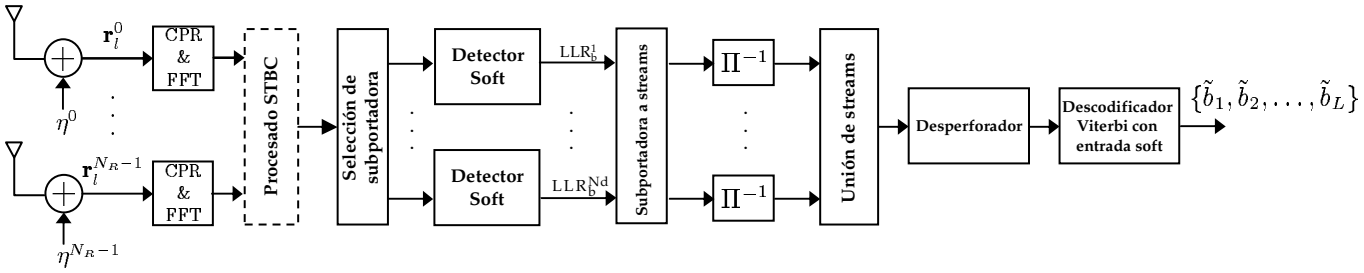


Figura 2. Diagrama de bloques de un receptor genérico IEEE 802.11n

$\boldsymbol{H}_{t+1}[k] = \boldsymbol{H}_t[k]$ la estimación MRC se puede expresar como[1]

$$\boldsymbol{y}^{STBC}[k] = \Gamma_k \boldsymbol{s}[k] + \tilde{\boldsymbol{\eta}}[k] \tag{6}$$

donde $\Gamma_k = \sum_{n_r=1}^{N_R} \left(|h_{1,n_r}[k]|^2 + |h_{2,n_r}[k]|^2 \right)$, con $h_{n_t,n_r}[k]$ denotando los coeficientes de canal entre la antena n_t y la antena n_r y $\tilde{\boldsymbol{\eta}}[k]$ es un vector AWGN de media cero y covarianza $E\{\tilde{\boldsymbol{\eta}}[k]\tilde{\boldsymbol{\eta}}[k]^H\} = \Gamma_k \sigma^2 \boldsymbol{I}_2$. La SNR de salida correspondiente al símbolo transmitido j se calcula como

$$SNR_j[k] = \frac{P_T \Gamma_k}{N_T \sigma^2}. \tag{7}$$

Alternativamente, si $N_s = 2$ se aplica un detector lineal MMSE sobre las muestras recibidas para separar los dos streams. Así, la estimación del símbolo MMSE viene dada por

$$\boldsymbol{y}^{SDM}[k] = \boldsymbol{W}[k]\boldsymbol{r}[k] \tag{8}$$

donde la matriz del filtro MMSE $\boldsymbol{W}[k]$ se calcula como

$$\boldsymbol{W}[k] = \left(\boldsymbol{H}^H[k]\boldsymbol{H}[k] + N_T \sigma^2 \boldsymbol{I}_{N_T} \right)^{-1} \boldsymbol{H}^H[k]. \tag{9}$$

La SNR post-MMSE del símbolo transmitido j es

$$SNR_j[k] = \frac{1}{\left[\left(\frac{P_T}{N_T \sigma^2} \boldsymbol{H}^H[k]\boldsymbol{H}[k] + \boldsymbol{I}_{N_T} \right)^{-1} \right]_{j,j}} - 1. \tag{10}$$

Después de obtener las expresiones para las estimaciones de cada uno de los símbolos transmitidos, tanto para STBC como para SDM (véanse (6) y (8), respectivamente), los símbolos estimados son transformados a información soft (LLR) para poder ser utilizados por el descodificador de Viterbi. Es conocido que la descodificación basada en información soft mejora considerablemente las prestaciones respecto a la descodificación hard. El proceso de generación de LLR sigue las aproximaciones de [13] donde la LLR para el bit en fase de la posición p-ésima del símbolo transmitido j se obtiene como

$$LLR(s_j[k], b_{I,p}[k]) = SNR_j[k] D_{I,p}[k] \tag{11}$$

siendo $D_{I,p}[k]$ definido en la Tabla I con $y_I[k] = \text{Re}\{y_j[k]\}$ donde $y_j[k]$ corresponde a los símbolos estimados de STBC o SDM según el modo MIMO utilizado y $SNR_j[k]$ corresponde a las expresiones (7) y (8) correspondientes a STBC y SDM, respectivamente. Las LLRs para los bits en cuadratura se calculan con un procedimiento análogo al usado para los bits en fase.

Modulación	$D_{I,1}[k]$	$D_{I,2}[k]$	$D_{I,3}[k]$
BPSK	$y_I[k]$		
QPSK	$y_I[k]$		
16QAM	$y_I[k]$	$-\lvert y_I[k]\rvert + 2$	
64QAM	$y_I[k]$	$-\lvert y_I[k]\rvert + 4$	$-\lvert\lvert y_I[k]\rvert - 4\rvert + 2$

Tabla I
TABLA AUXILIAR PARA EL CÁLCULO DE LLR.

III. ADAPTACIÓN RÁPIDA DEL ENLACE

Como se ha mencionado en la introducción, los algoritmos FLA intentan seleccionar el MCS que maximiza el *throughput* instantáneo sujeto a unas limitaciones de máxima probabilidad de *outage* de la PER, P_{out}, fijada una PER objetivo, PER_0. En consecuencia la capa física requiere una herramienta fiable para la predicción de la PER instantánea para cualquier modo MCS/MIMO posible, independiente de la longitud de paquete y/o realización de canal. Utilizando estas predicciones, se aplica una metodología de selección de modos MCS/MIMO en la capa MAC para asegurar el cumplimiento de los requisitos de QoS.

III-A. Predicción de la PER

Como se observa en (1), la PER se expresa como una función que depende del MCS $m \in \boldsymbol{\mathcal{M}}$, la SNR recibida, la longitud del paquete L y la realización de canal $\boldsymbol{H}$. Se han

[1]Para simplificar la notación y al ser las operaciones subsiguientes independientes del bloque de símbolos, a partir de ahora ya no se considerará el subíndice t.

propuesto diferentes aproximaciones en la literatura [14], [10] para asignar estos parámetros sobre una sola métrica de calidad (LQM) fácilmente asociable a un valor de PER por medio de tablas de referencia obtenidas a través de simulaciones *off-line*. De entre todas las estrategias de predicción propuestas, sólo aquellas basadas en una asignación unidimensional entre una LQM, conocida como SNR efectiva y la PER (véase, e.g., [14] y sus referencias) resultan particularmente atractivas. La SNR efectiva correspondiente a un MCS $m \in \boldsymbol{\mathcal{M}}$ se define como la SNR sobre el canal AWGN que requeriría el MCS *m* para obtener la misma PER que se obtiene en una realización del canal con selectividad frecuencial, pudiéndose calcular como

$$SNR_{eff}^{(m)} = \alpha_1^{(m)} J^{-1}\left(\frac{1}{N_s N_d}\sum_{j=1}^{N_s}\sum_{k=1}^{N_d} J\left(\frac{SNR_j[k]}{\alpha_2^{(m)}}\right)\right) \quad (12)$$

donde $J(\cdot)$ es una función LQM específica del modelo y $J^{-1}(\cdot)$ es su inversa. Los parámetros $\alpha_1^{(m)}$ y $\alpha_2^{(m)}$ permiten al modelo adaptarse a las características del MCS correspondiente. La métrica de SNR efectiva de capacidad (CESM) se obtiene con $J(\gamma) = \log_2(1+\gamma)$. La métrica de SNR efectiva exponencial (EESM) corresponde a $J(\gamma) = \exp(-\gamma)$. Otras métricas propuestas son, por ejemplo, métrica de SNR efectiva de la información mutua (MIESM) obtenida como $J(\gamma) = \mathcal{I}_m(\gamma)$, donde $\mathcal{I}_m(\cdot)$ representa la información mutua del MCS m, o la métrica de SNR efectiva logarítmica (LESM) caracterizada por $J(\gamma) = \log_{10}(\gamma)$.

Este artículo, sin pérdida de generalidad, se centra en la estrategia de predicción de PER basada en EESM. Los valores óptimos de los parámetros $\alpha_1^{(m)}$ y $\alpha_2^{(m)}$ en (12) se obtienen mediante una búsqueda exhaustiva dirigida a minimizar el error de estimación de la $SNR_{eff}^{(m)}$ promediada sobre un amplio conjunto de realizaciones de canal independientes $\boldsymbol{\mathcal{H}}$ y con un intervalo de tasa de error media $\boldsymbol{\mathcal{P}} = [PER_{\text{mín}}, PER_{\text{máx}}]$, resultando en,

$$(\alpha_{1_{\text{opt}}}^{(m)}, \alpha_{2_{\text{opt}}}^{(m)}) = \arg \min_{\alpha_1^{(m)}, \alpha_2^{(m)}} E_{\boldsymbol{\mathcal{H}},\boldsymbol{\mathcal{P}}}\left\{\left|SNR_{eff}^{(m)} - SNR_{AWGN}^{(m)}\right|^2\right\} \quad (13)$$

para todos los $m \in \boldsymbol{\mathcal{M}}$, donde $SNR_{AWGN}^{(m)}$ es la SNR requerida por el modo m para obtener una PER dada perteneciente a $\boldsymbol{\mathcal{P}}$ en el canal AWGN.

Para determinar $\alpha_1^{(m)}$ y $\alpha_2^{(m)}$, el sistema se calibra sobre un intervalo de variación de PER $\boldsymbol{\mathcal{P}} = [0{,}01 - 0{,}95]$ y sobre un conjunto $\boldsymbol{\mathcal{H}}$ de 200 realizaciones de canal obtenidas a partir de los perfiles de canal B y E, utilizando la herramienta de generación de canales MIMO descrita en [15]. Para mostrar el correcto funcionamiento del LQM, en la Fig. 3 se presentan las SNR_{eff} para cada realización de canal utilizando las $\alpha_1^{(m)}$ y $\alpha_2^{(m)}$ correspondientes. Para los MCS de un *stream* (Fig. 3a y Fig. 3b), el desplazamiento sobre la curva AWGN es casi ideal, las SNR_{eff} calculadas se aproximan muchísimo a la curva AWGN. Para dos *streams* (Fig. 3c y Fig. 3d) el desplazamiento no es tan preciso y se aprecian diferencias más pronunciadas, pero que resultan aceptables cuando se aplica AMC.

III-B. Proceso de selección del MCS

Para satisfacer las limitaciones de optimización correspondientes al outage de la PER, se requiere un umbral de SNR efectiva $SNR_{Th}^{(m)}$ para cada $m \in \boldsymbol{\mathcal{M}}$ de forma que

$$\Pr\left\{PER_{AWGN}^{(m)}\left(SNR_{eff}^{(m)}\right) > PER_0\right\} \leq P_{\text{out}} \quad (14)$$

siempre que $SNR_{eff}^{(m)} \geq SNR_{Th}^{(m)}$. La $SNR_{Th}^{(m)}$ se obtiene como la $SNR_{eff}^{(m)}$ que satisface (14) con igualdad. Esta probalidad es calculada numericamente utilizando todas las realizaciones del conjunto $\boldsymbol{\mathcal{H}}$. En este estudio, la PER objetivo es $PER_0 = 0{,}1$ y la probabilidad máxima de *outage* de la PER ha sido fijada en $P_{out} = 0{,}05$. En la Fig. 3 se representan, utilizando una línea discontinua vertical, los valores de $SNR_{Th}^{(m)}$ para los MCS correspondientes, asegurando así que el 95 % de las $SNR_{eff}^{(m)}$ obtenidas se encuentran por debajo de PER_0.

Para determinar el MCS óptimo de una realización de canal concreta, el algoritmo FLA determina la SNR efectiva para los diferentes MCSs ordenados de mayor a menor *throughtput*. Si la SNR efectiva del MCS evaluado está por debajo del nivel de $SNR_{Th}^{(m)}$, el MCS se marca como un posible modo para la transmisión, en caso contrario, el MCS se considera no apropiado. Este procedimiento iterativo continúa hasta que se selecciona un MCS o se han descartado todos, en tal caso, se selecciona el modo de no transmisión. Algunos *throughtputs* se pueden conseguir tanto mediante SDM como con STBC. Si ambos resultan válidos para la transmisión, se elige el MCS que utiliza STBC debido a su mayor eficiencia espectral para valores bajos de la SNR [11].

III-C. Predicción de PER basada en BER

Previamente se ha mencionado que los métodos de predicción de PER dependen de la longitud de cada paquete, implicando un procedimiento de calibración individual para cada L. Esto implica que para cada longitud de paquete se requerirán un conjunto de tablas de referencia. Para evitar esta situación, se presenta una nueva técnica FLA basada en BER, en lugar de PER, consiguiendo una estrategia independiente de L. La asunción principal para esta estrategia es que para paquetes lo suficientemente largos, la BER es independiente de L y en consecuencia, si BER y PER se pueden relacionar mediante una expression cerrada, la predicción de PER puede hacer-se en función de la BER.

La estimación basada en BER aplica la técnica EESM introducida anteriormente modificando algunas de sus características. Esta determina la $SNR_{eff}^{(m)}$ para cada realización de canal utilizando (12), con $\alpha_1^{(m)}$ y $\alpha_2^{(m)}$ obtenidos a partir de una calibración en que las curvas PER han sido substituidas por las curvas BER y los intervalos de PER se adaptan a BER utilizando la expresión que más adelante relaciona PER con BER. Además, se determina la $SNR_{Th}^{(m)}$ para cada $m \in \boldsymbol{\mathcal{M}}$ de forma que $\Pr\left\{BER_{AWGN}^{(m)}\left(SNR_{eff}^{(m)}\right) > BER_0^{(m)}\right\} \leq P_{\text{out}}$ siempre que $SNR_{eff}^{(m)} \geq SNR_{Th}^{(m)}$, dónde, como se presenta más adelante, la BER objetivo para cada MCS, llamada $BER_0^{(m)}$, se obtiene a partir de PER_0. El algoritmo de búsqueda del MCS de transmisión no requiere modificaciones.

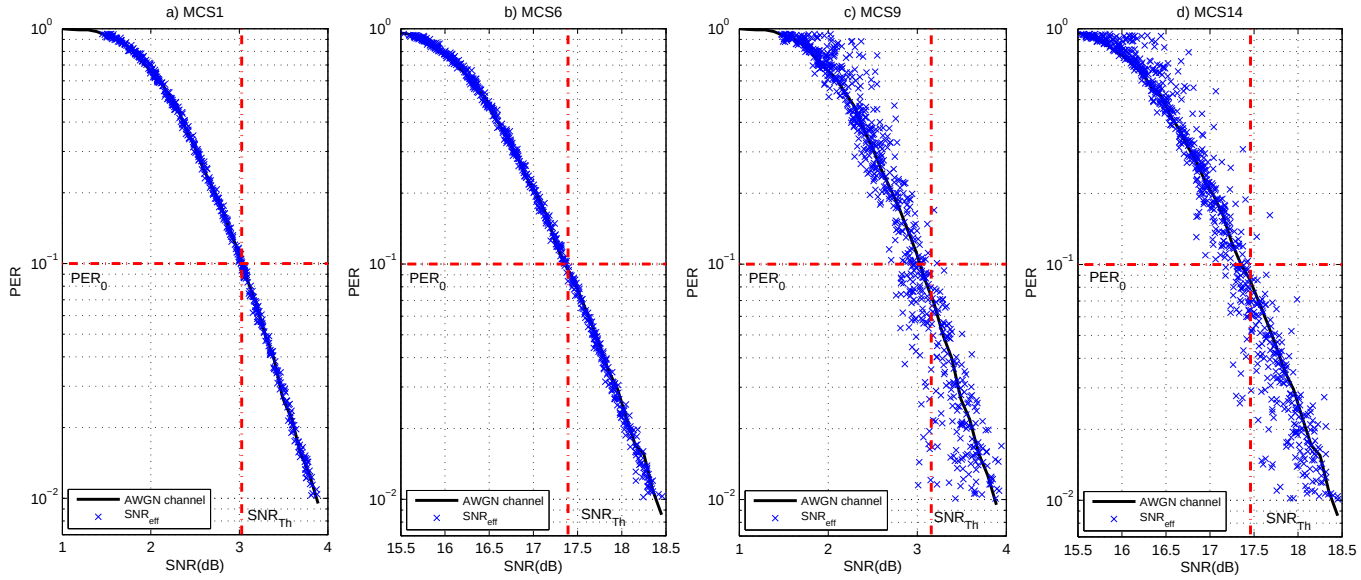


Figura 3. Desplazamiento de realizaciones sobre AWGN

La probabilidad de evento erróneo en un paquete codificado convolucionalmente utilizando el MCS m puede aproximarse como

$$P_e^{(m)} \approx P_b^{(m)}/d_f. \tag{15}$$

donde d_f es la distancia libre del código convolucional y $P_b^{(m)}$ es la probabilidad de error sin descodificar del MCS m. Esto se basa en la asunción que el número de bits erróneos por evento erróneo es aproximadamente igual a la distancia libre. Utilizando $P_e^{(m)}$, la PER del MCS m puede aproximarse como

$$PER^{(m)} \approx 1 - \left(1 - P_e^{(m)}\right)^{\left(\frac{L}{R_m}\right)}, \tag{16}$$

donde R_m es la tasa de codificación del MCS. Esta aproximación está basada en la suposición de que el paquete libre de errores es consecuencia de la ausencia de eventos erróneos sobre cada una de las posibles transiciones del diagrama trellis del código convolucional. Utilizando (15) y (16), $BER_0^{(m)}$ para cualquier $m \in \boldsymbol{\mathcal{M}}$ se puede obtener como una función de la PER objetivo PER_0, esto es,

$$BER_0^{(m)} = \left(1 - (1 - PER_0)^{\frac{R_m}{L}}\right) d_f. \tag{17}$$

En la Fig. 4 se compara la precisión de (15) y (16) utilizando la BER obtenida a partir del uso de paquetes de 1664 bits, para relacionar BER con PER respecto a los resultados reales de PER obtenidos para diferentes longitudes de paquetes. La Fig. 4 representa la transmisión de múltiples paquetes para una realización de canal concreta. Las curvas PER obtenidas por simulación se ordenan en prestaciones de menor a mayor longitud de paquete, siendo las curvas con longitudes más cortas las que obtienen valores de PER inferiores para la misma SNR. En cada MCS se observa que PER real y PER estimada no divergen demasiado y se puede dar como válida la aproximación. La sección IV proporciona los resultados de esta aproximación aplicados en el FLA, confirmando así la validez de esta técnica.

III-D. Error de estimación de canal

El proceso de estimación de canal utiliza los preámbulos y subportadoras piloto para estimar los coeficientes de la respuesta impulsional que minimizan el error cuadrático medio del error de estimación. Dado que las técnicas FLA consideradas asumen que el receptor dispone de información sobre el estado del canal, es importante considerar cómo afectan estas imperfecciones al proceso de adaptación. En el marco de este estudio se ha modelado el error de estimación utilizando un ruido Gaussiano blanco aditivo de media cero y varianza [16]

$$MSE = \frac{\tau_{max}}{T_{OFDM}}\sigma^2,$$

donde T_{OFDM} representa el período del símbolo OFDM y τ_{max} es el retardo máximo introducido por el canal. Nótese que el error de estimación es inversamente proporcional a la SNR.

IV. Resultados

El sistema simulado sigue las especificaciones del *draft* actual del IEEE 802.11n (véase Fig. 1) [8]. El sistema se configura para usar GI completo, $N_c = 64$ subportadoras sobre una banda de 20MHz con $N_d = 52$ portadoras de datos (las otras subportadoras son pilotos o nulas). Por otro lado y como ya se ha comentado en la Sección III, la PER objetivo ha sido fijada a $PER_0 = 0{,}1$ y $P_{out} = 0{,}05$. Los parámetros $\alpha_1^{(m)}$ y $\alpha_2^{(m)}$ se han determinado utilizando un conjunto de 200 realizaciones de canal a partir de los perfiles de canal B y E, obtenidas mediante la herramienta de generación de canales MIMO descrita en [15].

La figura 5a presenta los resultados del *throughput* que se obtienen utilizando estimación ideal en el canal B para los sistemas con MCS fijos, FLA y el algoritmo de límite de prestaciones (PBA). El PBA es un algoritmo FLA ideal que para cualquier realización de canal es capaz de seleccionar el MCS con el que se consigue el *throughput* máximo mientras se garantiza la transmisión libre de errores [9]. El FLA EESM

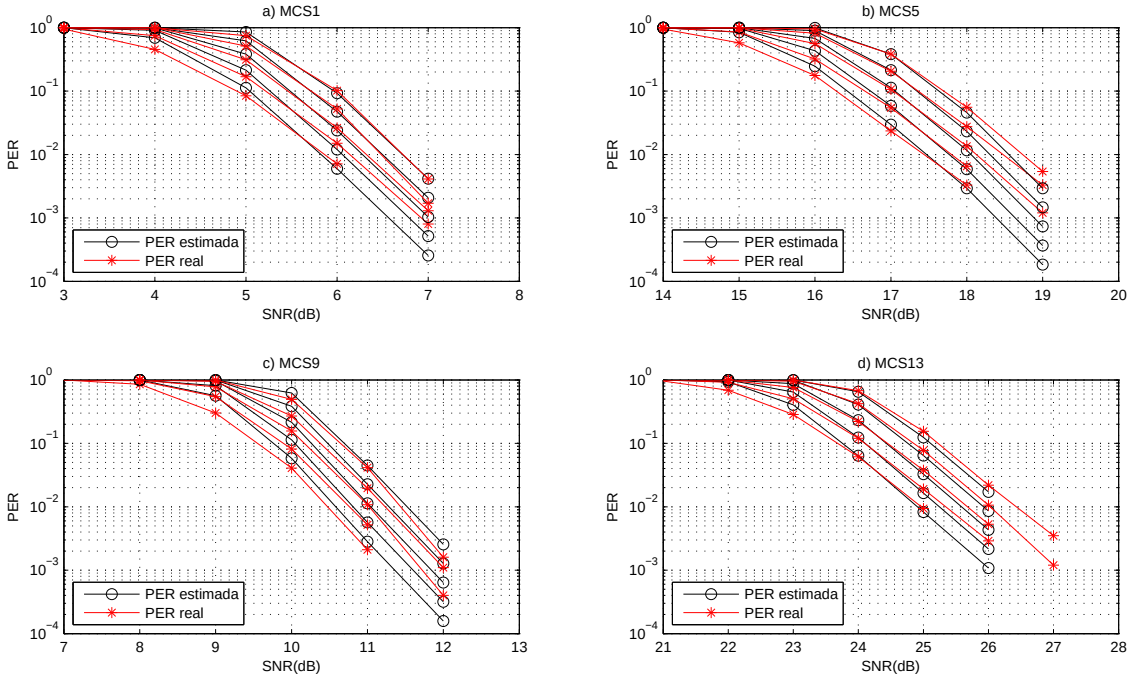


Figura 4. Resultados de PER a partir de la BER para L=416, 832, 1664 y 3328 para los MCSs: a)MCS1, b)MCS5, c)MCS9 y d)MCS13.

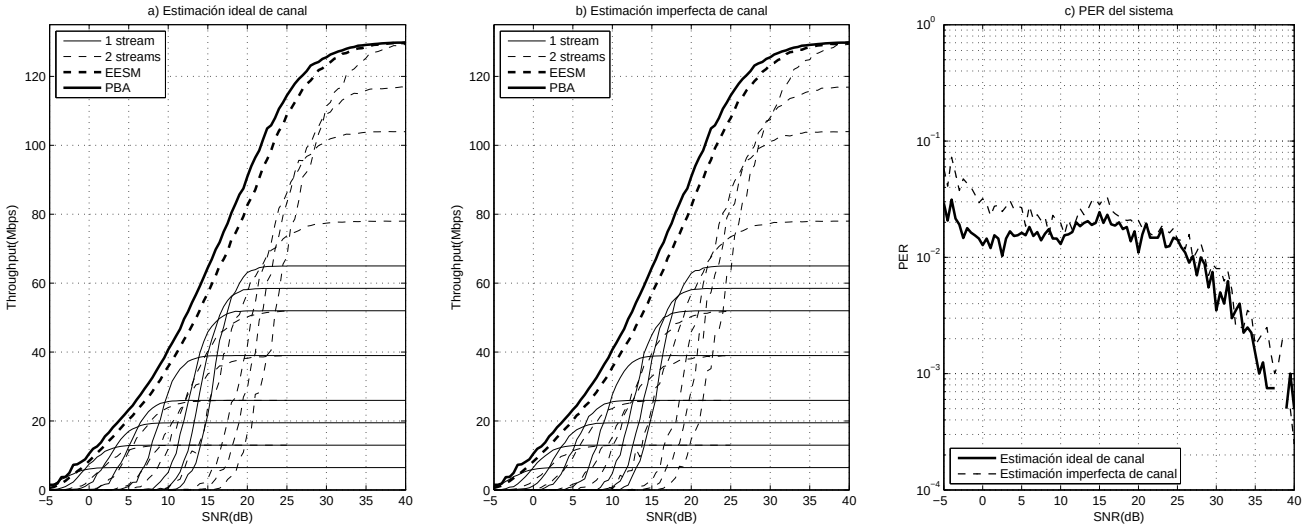


Figura 5. *Throughput* del sistema en el canal B: a) estimación ideal del canal, b) estimación imperfecta y c) PER para ambos sistemas.

claramente mejora las prestaciones de los MCS fijos y opera ligeramente por debajo de las prestaciones del PBA. Además, EESM cumple con las limitaciones de QoS manteniendo un valor de PER inferior a la PER objetivo PER_0 (Fig. 5c). Obviamente, la mayor diferencia entre la PER actual y la prescrita, resulta en un sistema demasiado pesimista, debido a la ausencia de control de potencia. Se obtienen resultados análogos para el canal E, presentados en las Figs. 6a y 6c, respectivamente. En este caso las ganancias entre sistemas fijos y adaptativos no son tan pronunciadas. Por otro lado, la PER sigue satisfaciendo los requisitos preestablecidos de QoS.

En las Fig. 5b y 6b se presentan los resultados obtenidos usando estimación imperfecta de canal. Para el canal B, los resultados son prácticamente idénticos a los obtenidos con estimación ideal. Esto se debe a que el error de estimación depende directamente del retardo máximo introducido por el canal y para el canal B ($80ns$) es poco significativo respecto a la duración de un símbolo OFDM ($3{,}2\mu s$). Como en el caso ideal, el sistema FLA también cumple con los requisitos de QoS preestablecidos inicialmente (véase Fig. 5). Por otro lado, en el canal E si que se observa una clara degradación de prestaciones, hasta el punto de empeorar el funcionamiento global del sistema entre 1 y 2 dBs. Además, el sistema

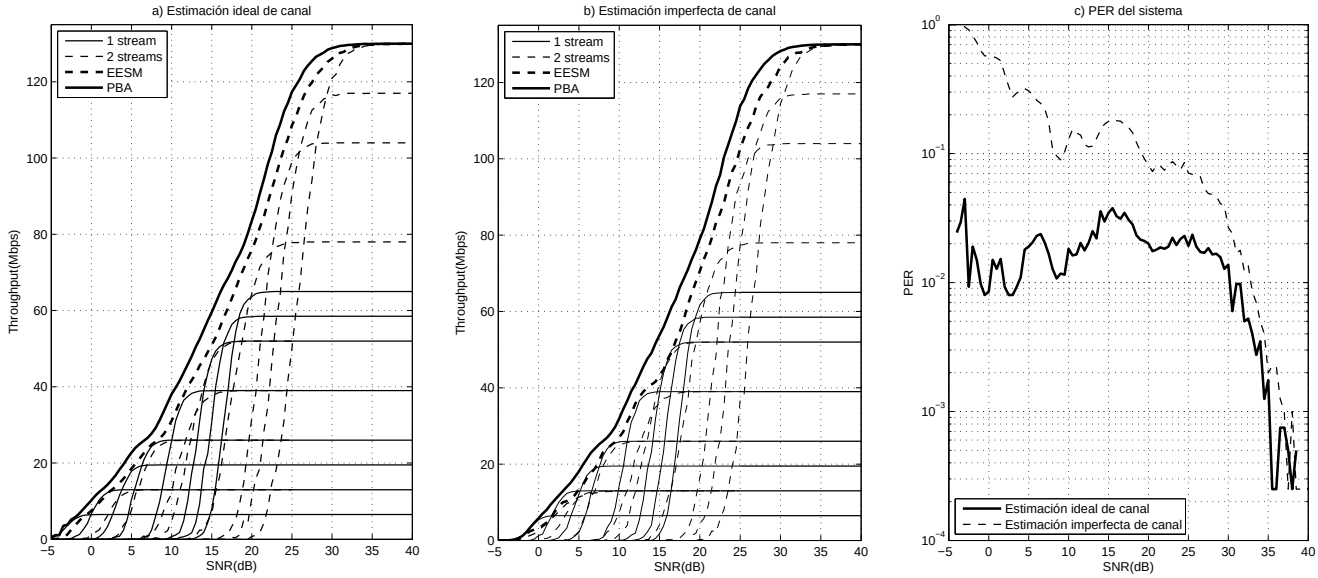


Figura 6. *Throughput* del sistema en el canal E: a) estimación ideal del canal, b) estimación imperfecta y c) PER para ambos sistemas.

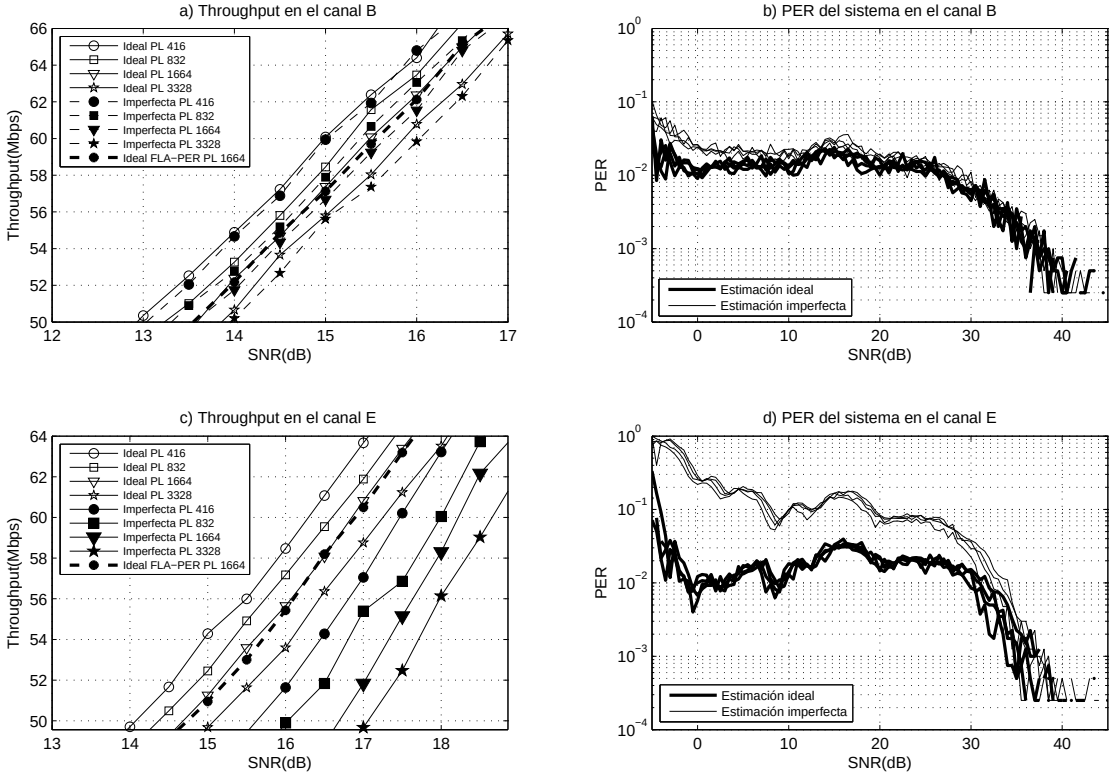


Figura 7. Prestaciones del FLA basado en BER: a) *Throughput* canal B, b) PER canal B, c) *Throughput* canal E y d) PER canal E.

no consigue satisfacer los requisitos de QoS preestablecidos (véase Fig. 6 c). En el canal E hay más diversidad temporal, introduciendo retardos superiores en el sistema ($720ns$) que conllevan una estimación de canal más imprecisa. Para prevenir el incumplimiento de los requisitos de QoS prescritos se tendrían que utilizar otras técnicas adaptativas, mejorar el sistema de estimación del canal o intentar reforzar el sistema de determinación de $SNR_{Th}^{(m)}$.

La Figura 7 presenta los resultados obtenidos utilizando métodos de predicción de PER basada en BER con la longitud de paquete como parámetro. El proceso de calibración se ha llevado a cabo utilizando realizaciones de prestaciones BER para un paquete de longitud $L \approx 1664$ bits. Estos resultados posteriormente se han extrapolado para predecir las prestaciones PER para longitudes de paquete entre $L = 416$ y $L = 3328$ bits. En el canal B (Fig. 7a), FLA-BER con estimación ideal de canal obtiene resultados idénticos al EESM basado en PER. Cuando el sistema utiliza paquetes más cortos sus prestaciones mejoran, siendo siempre la configuración con paquetes más cortos la que obtiene tasas de transmisión más altas para la misma SNR. Para cualquier longitud de paquete la PER del sistema es considerablemente inferior a PER_0, cumpliendo así la QoS requerida (véase 7b). Por otro lado, en el canal E (Fig. 7c y Fig. 7d), cuando hay errores en la estimación del canal, las prestaciones del algorismo FLA-BER decrecen al igual que sucedía en el sistema EESM basado en PER. El sistema reduce aproximadamente sus prestaciones entre 1 y 2 dBs respecto del sistema ideal y la PER del sistema no siempre se mantiene por debajo de PER_0. Este comportamiento se observa para cualquier longitud de paquete y se debe al efecto del error de estimación del canal, no al sistema FLA-BER utilizado.

V. Conclusiones

En este artículo se ha presentado una novedosa metodología para el diseño intercapas (*cross-layer*) de técnicas FLA *fast-link adaptation* para WLANs basadas en MIMO-OFDM. Esta metodología se ha aplicado al caso concreto de IEEE 802.11n, demostrando su capacidad para optimizar la eficiencia del sistema sujeta a restricciones de QoS en términos de probabilidad de *outage* de la PER. Los MCSs correspondientes a uno y dos *streams* han sido evaluados utilizando STBC y SDM, respectivamente. Se ha observado que FLA con predicción de PER basada en estimaciones ideales de canal satisface los requerimientos de PER obteniendo prestaciones sólo ligeramente inferiores a las obtenidas con un sistema ideal (PBA–*performance bound algorithm)*. Cuando hay errores en la estimación de canal, bien debido a una SNR operativa baja o a un canal con una selectividad frecuencial muy elevada, las prestaciones de los algoritmos FLA en términos de *throughput* se ven significadamente afectadas. Por ejemplo, para el modelo de canal E, con un retardo máximo de 720ns, el error de estimación de canal es muy elevado y la PER instantánea ocasionalmente supera los objetivos de QoS. En cambio, para el canal B, con un retardo máximo de 80ns, el error de estimación no es tan pronunciado y el sistema consigue resultados similares a los obtenidos con una estimación ideal del canal. Cabe destacar, también, que se ha introducido una nueva variante del algoritmo FLA basada en la predicción de BER (en lugar de PER) utilizando EESM. Esta técnica obtiene resultados casi idénticos a los métodos análogos basados en PER reduciendo el costoso procedimiento de calibración/predicción.

Agradecimientos

Esta investigación ha sido parcialmente financiada por el MEC y FEDER en el marco del proyecto COSMOS (TEC2008-02422), Conselleria d'Economia, Hisenda i Innovació del Govern de les Illes Balears en el marco de los proyectos XISPES (PROGECIB-23A), PCTIB-2005GC1-09 y beca predoctoral, y una beca Ramón y Cajal del Gobierno español (co-financiada por el Fondo Social Europeo).

Referencias

[1] A. Kamerman and L. Monteban, "WaveLAN-II: a high-performance wireless LAN for the unlicensed band," *Bell Labs Technical Journal*, vol. 2, no. 3, pp. 118–133, 1997.

[2] J. Jelitto, A. N. Barreto, and L. H. Truong, "Link adaptation," Patent No. WO/2004/004194, Jan. 2004.

[3] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: a practical approach," in *Proc. ACM MSWIN'04*, 2004, pp. 126–134.

[4] D. Qiao and S. Choi, "Fast-responsive link adaptation for IEEE 802.11 WLANs," in *Proc. IEEE ICC'05*, vol. 5, 2005, pp. 3583–3588.

[5] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *Proc. MobiCom '06*, 2006, pp. 146–157.

[6] J. Kim, S. Kim, S. Choi, and D. Qiao, "CARA: Collision-aware rate adaptation for IEEE 802.11 WLANs," in *Proc. IEEE INFOCOM'06*, 2006, pp. 1–11.

[7] W. H. Xi, A. Munro, and M. Burton, "Link adaptation algorithm for the IEEE 802.11n MIMO system," in *Proc. Networking'08, LNCS 4982*, 2008, pp. 780–791.

[8] S. A. Mujtaba, "TGn sync proposal technical specification. doc.: IEEE 802.11-04/0889r7," Draft proposal, July 2005.

[9] T. Jensen, S. Kant, J. Wehinger, and B. Fleury, "Mutual information metrics for fast link adaptation in IEEE 802.11n," *ICC*, pp. 4910–4915, May 2008.

[10] S. Simoens, S. Rouquette-Léveil, P. Sartori, Y. Blankenship, and B. Classon, "Error prediction for adaptive modulation and coding in multiple-antenna OFDM systems," *Elsevier Signal Process.*, vol. 86, no. 8, pp. 1911–1919, 2006.

[11] Y.-S. Choi and S. Alamouti, "A pragmatic PHY abstraction technique for link adaptation and MIMO switching," *IEEE JSAC*, vol. 26, no. 6, pp. 960–971, August 2008.

[12] G. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Technical Journal*, vol. 1, no. 2, pp. 41–59, 1996.

[13] F. Tosato and P. Bisaglia, "Simplified soft-output demapper for binary interleaved COFDM with application to HIPERLAN/2," *ICC*, vol. 2, pp. 664–668, 2002.

[14] K. Brueninghaus, D. Astely, T. Salzer, S. Visuri, A. Alexiou, S. Karger, and G.-A. Seraji, "Link performance models for system level simulations of broadband radio access systems," *PIMRC*, pp. 2306–2311, Sept. 2005.

[15] J. Kermoal, L. Schumacher, K. Pedersen, P. Mogensen, and F. Frederiksen, "A stochastic MIMO radio channel model with experimental validation," *IEEE JSAC*, vol. 20, no. 6, pp. 1211–1226, Aug 2002.

[16] Y. Li, "Simplified channel estimation for OFDM systems with multiple transmit antennas," *Wireless Communications, IEEE Transactions on*, vol. 1, no. 1, pp. 67–75, Jan 2002.

Hacia la norma IEEE 802.11n: Caracterización experimental de las extensiones de capa MAC para la mejora del rendimiento en redes WLAN

D. Gómez, R. Agüero, M. García, R. Sanz, L. Muñoz
Departamento de Ingeniería de Comunicaciones
Universidad de Cantabria
Laboratorios de I+D de Telecomunicaciones
Plaza de la Ciencia, Avda de los Castros
39005-Santander, SPAIN
E-mail: david.gomez@alumnos.unican.es, {ramon, marta, roberto, luis}@tlmat.unican.es

***Resumen*- En este trabajo se analiza, de manera completamente experimental, el comportamiento de las extensiones que se han venido añadiendo al protocolo de acceso al medio empleado por la tecnología IEEE 802.11 para mejorar sus prestaciones y que, además, se van incorporando paulatinamente al propio estándar, en las diferentes ampliaciones que al mismo se están realizando, como es el caso, por ejemplo, de la norma IEEE 802.11n (que se espera sea publicada el año que viene). Concretamente, este estudio se centra en la caracterización experimental de dos técnicas propietarias del fabricante Atheros, denominadas *Bursting*, que se refiere a la capacidad de transmitir tramas de manera consecutiva sin necesidad de contender por el canal, y *Fast Frames*, que concatena tramas antes de transmitirlas, para reducir la sobrecarga correspondiente. Además se tienen en cuenta mecanismos de QoS, definidos en la extensión IEEE 802.11e. Dicha caracterización se ha llevado a cabo sobre las especificaciones de capa física 802.11b y 802.11 g, fijando la tasa binaria a 11 y 54 Mbps, respectivamente. El trabajo se complementa con un amplio estudio teórico de las prestaciones de las diferentes técnicas, lo que permite contrastar la validez de los resultados obtenidos experimentalmente.**

Palabras Clave- 802.11n, Rendimiento, WLAN, Fast Frames, Bursting

I. INTRODUCCIÓN

La presencia de la tecnología inalámbrica basada en el estándar IEEE 802.11 es, a día de hoy, una realidad, como se desprende del hecho de que las ventas de dispositivos portátiles hayan aumentado en los últimos años a un ritmo muy notable. Este crecimiento no sólo se pone de manifiesto con las cifras del mercado, sino que, del mismo modo, la propia tecnología también avanza a un ritmo vertiginoso. Así, y aunque en el momento de su estandarización, las capacidades de estas redes pudiesen parecer suficientes para las aplicaciones que se utilizaban en aquellos tiempos, como podían ser el correo electrónico o la navegación web, ha surgido recientemente una serie de nuevos servicios que requieren de una capacidad sensiblemente mayor (por ejemplo, la televisión por Internet) y que además, requieren de unos niveles de calidad de servicio, QoS (retardo medio, jitter, throughput…) que no fueron previstos en los planteamientos iniciales del estándar.

En este marco, el objetivo de este documento es el de analizar experimentalmente el comportamiento de esta tecnología en unas condiciones cercanas a las ideales, lo que dará una idea de sus límites máximos. En concreto se analizarán las dos recomendaciones físicas con mayor presencia en la actualidad (802.11b y 802.11g), sobre las que se estudiará cuál es el beneficio adicional que ciertas extensiones de la capa MAC, incluidas ya en alguna de las ampliaciones del estándar 802.11, pueden aportar, en la misma línea que otros trabajos previos [1-4], la mayoría de ellos basados en simulaciones.

Para ello, el artículo se ha estructurado como sigue: en la Sección II se introducen las ampliaciones al estándar que se caracterizarán, cuyo rendimiento teórico se estudia en la Sección III. Posteriormente, las Secciones IV y V describen la plataforma de medidas empleada y presentan los resultados obtenidos, respectivamente. Finalmente, en la Sección VI se concluye el trabajo, proponiendo un conjunto de aspectos que quedan abiertos a raíz de la investigación llevada a cabo.

II. AMPLIACIONES AL ESTÁNDAR IEEE 802.11

El estándar IEEE 802.11 establece como esquema de acceso al medio por defecto el método DCF (Distributed Coordination Function), función básica que proporciona un servicio de tipo "best-effort". Se trata de un mecanismo distribuido que se basa en el protocolo CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance). Cuando una estación quiere transmitir una trama, monitoriza la actividad del canal; si lo encuentra libre durante un intervalo de tiempo denominado DIFS (Distributed InterFrame Space), la estación transmite. Por el contrario, si el medio está ocupado (ya sea inmediatamente o después de esperar el DIFS), la estación continúa monitorizando el canal hasta que lo encuentre libre durante un DIFS. En ese momento, la estación deberá inicializar un temporizador y esperar un número aleatorio de ranuras temporales antes de comenzar con su transmisión. Dicho temporizador se va reduciendo en tanto en cuanto el canal siga libre, congelándose cuando se ocupe y volviéndose

a reactivar al liberarse el canal de nuevo durante un DIFS. Cuando el temporizador llega a cero, la estación puede iniciar la transmisión. El número de ranuras o slots de *backoff* a esperar es una variable aleatoria uniformemente distribuida en el intervalo [0, CW-1], donde CW es la ventana de contienda. En el primer intento de transmisión, CW=CWmin y este parámetro se dobla cada vez que se retransmite la trama hasta alcanzar el valor máximo, CWmax. Dichos límites, así como la duración de la ranura, dependen de la especificación de capa física utilizada.

Tras la correcta recepción de una trama, el receptor debe iniciar la transmisión del correspondiente reconocimiento, después de esperar un tiempo llamado Short InterFrame Space (SIFS), que debe ser menor que el DIFS para evitar que otra estación comience a transmitir información antes de la recepción del ACK en el transmisor. Una vez recibido éste, si la estación transmisora tiene más tramas por enviar deberá esperar un tiempo aleatorio de *backoff*, incluso si el medio se encuentra libre después de un DIFS, para evitar la captura del canal. Por otro lado, si la estación transmisora no recibe el ACK transcurrido un tiempo dado, la trama es retransmitida de acuerdo con las reglas establecidas en el proceso de *backoff* exponencial binario descrito anteriormente. Notar que la ausencia de un ACK puede ser debida tanto a una colisión como a una condición de error en el canal. La Figura 1.a muestra el intercambio de tramas que se acaba de describir para el modo de funcionamiento básico (DCF).

En noviembre de 2005 se aprueba la norma IEEE 802.11e para el soporte de Calidad de Servicio o QoS (Quality of Service) en las redes WLAN [5]. Este estándar introduce una nueva función denominada HCF (Hybrid Coordination Function), destinada a permitir a las estaciones mantener múltiples colas de servicio y garantizar un acceso priorizado al medio inalámbrico para los nodos que requieran mejor calidad de servicio. Para ello, define un nuevo método de acceso denominado EDCA (Enhanced Distributed Channel Access), mecanismo distribuido y basado en contienda, sucesor del DCF. Cada estación distingue hasta cuatro categorías de acceso (AC, Access Category) según se trate de tráfico de voz, vídeo, best effort y background, estableciendo un valor diferente de CWmin para cada una de ellas. Adicionalmente, el esquema EDCA especifica nuevas reglas de utilización del canal, que se basan en los principios de "Transmisión Múltiple de Tramas durante una Oportunidad de Transmisión" (TXOP, Transmission Opportunity) y "Reconocimiento de bloque" o Block ACK. El primero de ellos permite que una estación que consiga el acceso al canal pueda transmitir una o más tramas, (pertenecientes a la misma categoría de acceso) en una ráfaga, separadas únicamente por un tiempo SIFS. La duración máxima de la ráfaga no debe superar un umbral denominado TXOP limit. Por otra parte, el mecanismo de Block ACK permite que un bloque de tramas pueda ser confirmado por un único ACK. Mencionar que dicho bloque puede enviarse en varios TXOPs.

Por otra parte, el draft 9.0 del futuro estándar IEEE 802.11n añade nuevas propuestas con objeto de seguir mejorando la eficiencia de la capa MAC, a costa de reducir la sobrecarga asociada [6]. Una de ellas, denominada A-MSDU (Aggregation MSDU), consiste en la agregación de varias unidades de datos de servicio en una única de unidad de datos de protocolo de capa MAC (MPDU), es decir, varios paquetes de capas superiores agrupados en una sola trama, con una única cabecera MAC 802.11 común. Adicionalmente, proponen otro esquema de agregación, A-MPDU (Aggregation MPDU), que consiste en que varias MPDUs compartan una misma cabecera física (pero manteniendo las diferentes cabeceras MAC). Al igual que en la norma 802.11e, se sigue contemplando y expandiendo el mecanismo de envío de tramas en ráfagas así como el esquema de Block ACK.

Algunas de estas mejoras han sido ya implementadas en soluciones propietarias desarrolladas por los fabricantes, como es el caso de las técnicas de *Bursting* y *Fast Frames* de los adaptadores inalámbricos Proxim de Atheros. La primera de ellas trata de incrementar el throughput efectivo en una transmisión 802.11, reduciendo la sobrecarga introducida por el mecanismo de acceso al medio con un planteamiento similar al EDCA TXOP de 802.11e. En el acceso por defecto que establece el estándar, si una estación quiere transmitir varias tramas de manera consecutiva, debe esperar a que el medio esté libre durante un tiempo DIFS más el número de ranuras que haya obtenido aleatoriamente comprendido entre 0 y (CWmin – 1). Gracias al modo *Bursting*, un elemento de la red puede apropiarse del canal y enviar tramas en ráfaga, es decir, la separación entre tramas será únicamente de un intervalo SIFS, en lugar de esperar el tiempo DIFS más el backoff del modo normal. Con esto se asegura que ninguna estación pueda detectar libre el medio e interrumpir la transmisión. La Figura 1.b representa el intercambio de tramas en el modo *Bursting*.

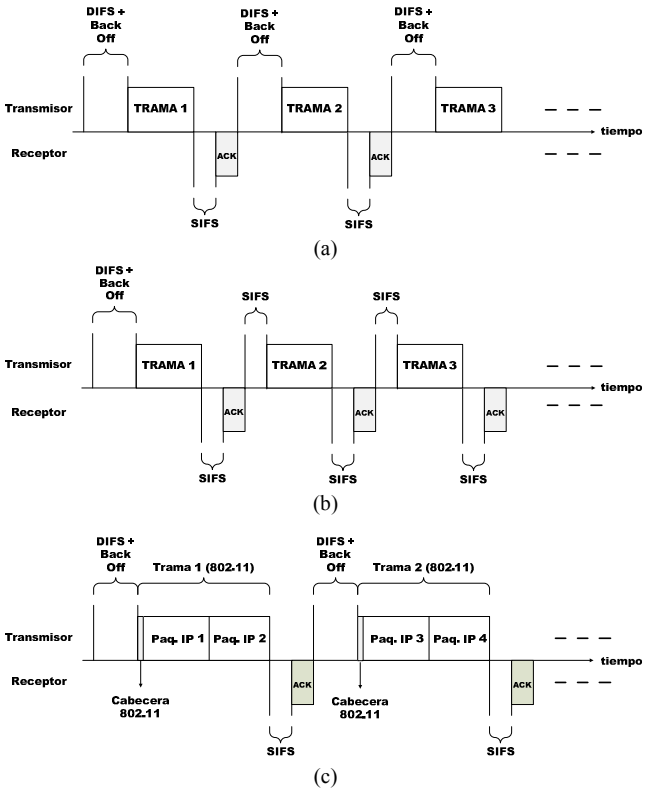


Fig.1. Acceso al medio en el modo básico (a), *Bursting* (b) y *Fast Frames* (c)

Para evitar que el equipo se apodere del canal de forma indefinida, se establece un límite de tiempo en el que enviar ráfagas, establecido por el parámetro TXOP limit de 802.11e.

La segunda técnica, denominada *Fast Frames*, permite encadenar 2 paquetes en una sola trama, con una única cabecera MAC 802.11, siguiendo una estrategia similar a la

propuesta de agregación A-MSDU del draft de 802.11n [7]. Así se consigue una mejora importante en el rendimiento global del sistema, dado que el segundo paquete no tiene que contender por el acceso al canal, reduciendo parte de la sobrecarga que introducen las capas física y MAC. La Figura 1.c muestra el intercambio de tramas en el modo *Fast Frames*, para el caso de tráfico IP.

III. Análisis del Rendimiento y Retardo Medio

Antes de presentar los resultados experimentales obtenidos sobre la plataforma de medidas que será descrita en la siguiente sección, es conveniente tener una idea previa de los valores que se van a obtener. Por esta razón, en este apartado, se realizará una estimación teórica de los límites de las extensiones de capa física (b y g), en términos de su rendimiento y retardo asociados.

Para obtener el rendimiento, en Mbps, se calculará el retardo medio entre las tramas que alcanzan el receptor, obteniéndose la tasa binaria de datos con la siguiente expresión:

$$Thput\ (Mbps) = \frac{Longitud_{Info}(bits)}{\overline{Retardo}\ (\mu seg)} \quad (1)$$

Como lo que se desea es conocer los límites de la capacidad de transmisión, se ha optado por ajustar el tamaño de las tramas a la MTU (Maximum Transfer Unit) de Ethernet, con 1500 bytes de carga útil o *payload*. Dado que los experimentos se realizarán con tráfico UDP/IP, cada datagrama transportará, por tanto, 1472 bytes de datos. En el modo *Fast Frames*, cada trama encadena dos datagramas, por lo que se duplica el número de bits de información por trama.

Según se puede observar en la Figura 1 (a), el retardo medio entre tramas, para la especificación 802.11b, se calcula como la suma de todas las contribuciones de las capas física y MAC, aplicando la expresión (2):

$$\overline{Retardo} = DIFS + \overline{BO} + t_{PLCP} + (t_{trans})^{DATOS} + \\ + SIFS + t_{PLCP} + (t_{trans})^{ACK} \quad (2)$$

Dado que el proceso de *Backoff* sigue una distribución uniforme con rango [0, CW_{min} – 1], su valor medio se obtendrá de acuerdo con la expresión (3):

$$\overline{BO} = \frac{(CW_{min} - 1)}{2} \cdot Slot_{time} \quad (3)$$

La Tabla 1 recoge los valores que especifica la norma 802.11b para los parámetros anteriores. Se utiliza el formato corto de preámbulo y cabecera PLCP (Physical Layer Convergence Procedure).

Tabla 1. Parámetros característicos de la norma IEEE 802.11b

Parámetro	*Valor*
Slot time	20 µseg
SIFS	10 µseg
DIFS = 2 · Slot time + SIFS	50 µseg
t_{PCLP} (Formato corto)	96 µseg
CW_{min} (Sin WMM)	32 slots
CW_{min} (Con WMM)	16 slots
R_{datos}	11 Mbps
R_{ACK}	2 Mbps

En el caso de la especificación de capa física 802.11g, el retardo medio se obtiene de forma similar, aplicando los parámetros de la Tabla 2 y añadiendo además la contribución debida a la compatibilidad con la norma b (*Signal Extension*).

Las medidas realizadas se clasificarán atendiendo a seis grupos: modo *puro*, *WMM (Wi-Fi MultiMedia)*[1], *Fast Frames*, *Bursting* (estos cuatro para comprobar su trabajo de forma independiente), *Fast Frames+Bursting* y por último *WMM+Fast Frames+Bursting*, con el fin de ver si se consiguen mejoras cuando se utilizan dichas técnicas de forma simultánea.

Tabla 2. Parámetros característicos de la norma IEEE 802.11g

Parámetro	*Valor*
Slot time	9 µseg
SIFS	10 µseg
DIFS = 2 · Slot time + SIFS	28 µseg
t_{Preamb}	16 µseg
t_{PLCP}	4 µseg
Signal extension	6 µseg
CW_{min} (Sin WMM)	16 slots
CW_{min} (Con WMM)	8 slots
R_{datos}	54 Mbps
R_{ACK}	24 Mbps

En el modo *Bursting* la primera trama se transmite de manera normal, pero las siguientes serán emitidas en forma de ráfaga hasta que se complete el tiempo TXOP que define el estándar 802.11e (valor fijado por defecto a 2048 µseg). Para estimar el número de tramas que se concatenarán para formar la ráfaga, basta con dividir el valor del TXOP limit entre el tiempo de transmisión de una trama (sin contar el proceso de Backoff), tomando el entero inferior:

$$Ráfaga = \left\lfloor \frac{TXOP}{\tau} \right\rfloor \quad (4)$$

siendo el valor de τ:

$$\tau = t_{PLCP} + (t_{trans})^{DATOS} + SIFS + t_{PLCP} + \\ + (t_{trans})^{ACK} = 1375\ \mu seg \quad (5)$$

Para los valores de la especificación 802.11b, la ráfaga sólo consta de una trama por lo que la mitad de ellas no tendrá que esperar, mientras que el resto elegirá un tiempo de backoff siguiendo la distribución del modo puro. Por otra parte, aplicando los parámetros de la norma 802.11g se deduce que el valor de τ es de 294 µseg, lo que supone el envío de ráfagas de 6 tramas.

Las Tablas 3 y 4 muestran los valores teóricos obtenidos para el retardo medio y para el rendimiento junto con el porcentaje de mejora respecto del modo básico. Por otra parte, las Figuras 2 y 3 muestran las funciones densidad de probabilidad (fdp) para cada una de las configuraciones analizadas.

Analizando la Tabla 3 se puede observar que, mediante el empleo de técnicas adicionales al modo básico DCF en 802.11b se logra mejorar el rendimiento en un porcentaje importante, con lo que a falta de otros factores (errores en el canal, colisiones, relación señal a ruido…) que se verán en el apartado de las medidas de campo, parece importante su utilización. En cuanto a la norma 802.11g (Tabla 4), se observa que la mejora que se alcanza es aún más relevante, sobre todo con el modo *Bursting* activado.

En el modo *Fast Frames*, al duplicar el tamaño de datos por trama, la sobrecarga se reduce con lo que se mejora

[1] Soporte de calidad de servicio, siguiendo la norma 802.11e. En este trabajo sólo se utiliza la categoría AC correspondiente al tráfico de tipo *best effort*, con CW_{min} igual a 16 slots para 802.11b y 8 para 802.11g.

notablemente el rendimiento. Lógicamente la distribución de los tiempos entre tramas (Figuras 2.d y 3.d) sigue siendo la del modo puro, pero con valores superiores, por ser las tramas de mayor longitud. Destacar que, mientras que, en el caso de 802.11g la combinación de *Fast Frames+Bursting* ofrece mayores prestaciones que cualquiera de estas dos técnicas por individual, esto no se cumple en 802.11b, dado que el TXOP limita el tamaño de la ráfaga (al ser la longitud de las tramas la especificada por *Fast Frames*). Concretamente, en el caso de IEEE 802.11g se pueden llegar a transmitir hasta 4 *super-tramas* por ráfaga, lo que se traduce en un incremento de más del 15% en el rendimiento respecto del modo Fast Frames individual.

Tabla 3. Resumen de valores teóricos para la norma 802.11b

Técnica	*Bytes datos*	*Retardo medio (µseg)*	*Thput (Mbps)*	*Mejora (%)*
Puro	1472	1735	6.78	0
WMM	1472	1575	7.48	10.32
Bursting	1472	1560	7.54	11.21
Fast Frames	2944	2838	8.29	22.27
FF + Burst	1472	1560	7.54	11.21
WMM+FF+Burst	1472	1480	7.95	17.26

Tabla 4. Resumen de valores teóricos para la norma 802.11g

Técnica	*Bytes datos*	*Retardo medio (µseg)*	*Thput (Mbps)*	*Mejora (%)*
Puro	1472	390.16	30.18	0
WMM	1472	354.16	33.25	10.17
Bursting	1472	316	37.26	23.46
Fast Frames	2944	607	38.74	28.36
FF + Burst	2944	543.72	43.31	43.51
WMM+FF+Burst	2944	534.72	44.04	45.92

Finalmente, mencionar que el efecto del *WMM* es simplemente apreciable en el ancho de la fdp del retardo correspondiente, ya que únicamente se limita el tamaño de la ventana de contención mínima.

IV. PLATAFORMA DE MEDIDAS

A. Configuración

Para la caracterización en un escenario real del canal inalámbrico, se ha planteado la creación de una celda WLAN formada por dos equipos situados a una distancia que puede considerarse nula, para así evitar que se pierdan tramas debido a las imperfecciones del canal, con una configuración de *hostapd*, donde uno de ellos actuará como punto de acceso, mientras que el otro lo hará como estación (modo infraestructura).

Se ha tratado de emplazarlos en un lugar óptimo desde el punto de vista de evitar en la medida de lo posible interferencias de otros puntos de acceso, estaciones y demás equipos que afecten a las medidas realizadas (esta condición junto con la de tener distancia cero, hacen que el comportamiento del canal pueda equipararse al caso ideal).

Como interfaz inalámbrica se han elegido las tarjetas de red "Proxim Orinoco Gold a/b/g Combo Card", las cuales incorporan un chipset de Atheros, por lo que se ha utilizado el driver desarrollado por MadWifi para Linux, el cual, al ser código abierto, permite su libre modificación, pues ello es necesario para implementar la configuración adecuada del entorno de medidas.

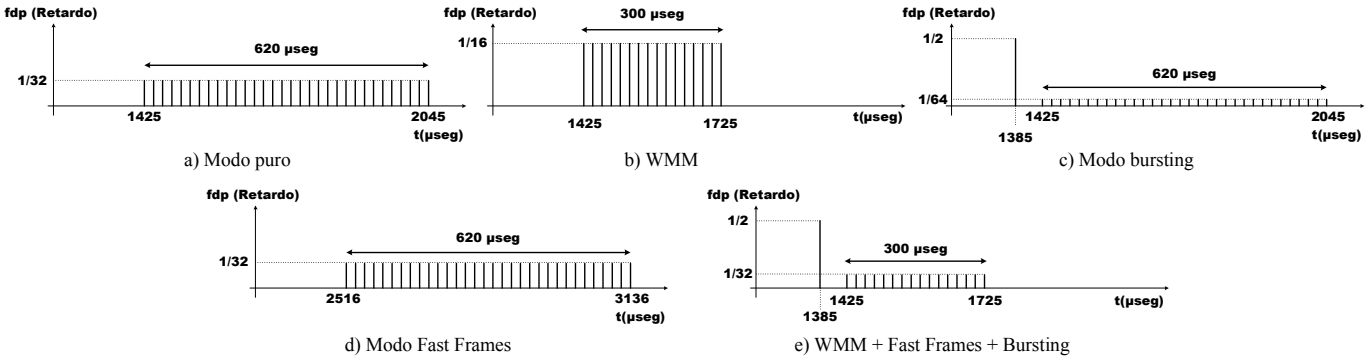


Fig. 2. Fdp de los retardos entre tramas teóricos para la recomendación física IEEE 802.11b

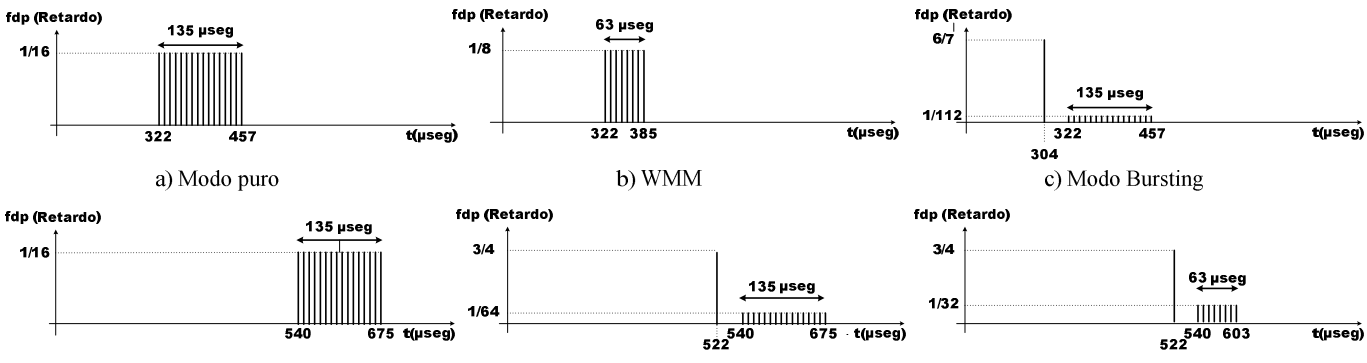


Fig. 3. Fdp de los retardos entre tramas teóricos para la recomendación física IEEE 802.11g

El esquema que se ha empleado es el que se muestra en la Figura 4: el equipo que hace las veces de AP será el encargado de transmitir tráfico UDP de manera continuada (recordar que son pruebas para estimar la capacidad límite del enlace), mientras que la estación receptora se encargará de monitorizar la comunicación, extraer información y datos de la misma, para procesarlos, devolviendo valores estadísticos y gráficas que caracterizarán el comportamiento de la comunicación.

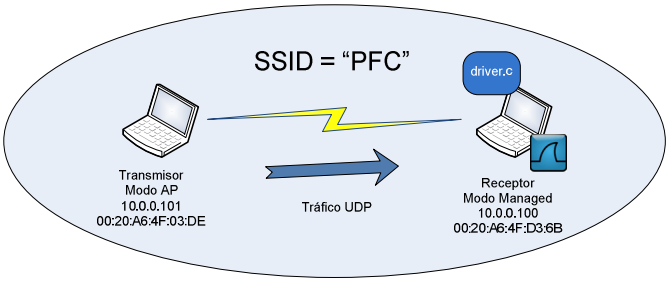


Fig. 4. Esquema del escenario de medidas

Al utilizar tráfico UDP se consigue que, por encima del nivel 802.11, no haya ningún tipo de control de flujo o errores, lo que permite caracterizar únicamente la capa de enlace. Esto, sumado a que se deshabilitarán las retransmisiones 802.11, supone que en el caso de producirse cualquier tipo de error en el envío, la trama se perderá.

De esta manera se consigue aislar el posible efecto del mecanismo de retransmisión que incorpora IEEE 802.11 que, en caso de estar activado, conseguiría reducir la pérdida IP a valores prácticamente nulos, con el consecuente incremento del rendimiento, incluso limitando el número de retransmisiones a una o dos tramas.

B. Herramientas empleadas

- Driver MadWifi de Atheros

Con el fin de obtener datos de las tramas recibidas (dirección MAC origen, relación señal a ruido o SNR, valor del campo CRC…) se modificará el código del driver para que se pueda acceder a esta información desde un programa de usuario, que será el encargado de volcarlo a ficheros traza que se procesarán posteriormente.

Para evitar incluir tramas no deseadas, se realizará un filtrado que garantice que solamente se recojan aquellas que provengan de la fuente deseada y, además, sean de datos (no de gestión o de control).

Por último, con objeto de evitar las retransmisiones inherentes a la capa MAC, el número de transmisiones por trama a nivel 802.11 se fijará a 1, con lo que cualquier error desembocará en la pérdida definitiva de esa información.

- Wireshark

Para completar los parámetros necesarios para caracterizar el canal, hacen falta otros valores adicionales, como el tiempo de recepción de las tramas o el campo *Identification* de la cabecera IP, que se obtendrán realizando capturas de la transmisión mediante este analizador de protocolos.

- Nttcp

Será el encargado de generar el tráfico UDP a enviar entre ambos equipos. Como se ha comentado en el apartado anterior, utilizará tramas con 1472 bytes de información (máximo valor posible sin fragmentar) y las enviará sin ningún tipo de espera adicional a la que se introduce con el mecanismo de acceso al medio del estándar 802.11.

- Matlab

Recibirá como parámetros de entrada los ficheros creados a partir de los datos del driver y las capturas de Wireshark, devolviendo los resultados que se analizarán en el siguiente punto, correspondiente a cada una de las medidas realizadas.

C. Procedimiento de medida

Para llevar a cabo todo el proceso de caracterización del comportamiento del canal se realizarán, para cada de las técnicas anteriormente mencionadas, una serie de 5 transmisiones de 25000 datagramas UDP cada una.

El esquema es el siguiente: el equipo AP actuará de transmisor NTTCP, mientras que la estación (receptor), usa el driver modificado para capturar los datos y el Wireshark monitorizando la medida.

En este documento se presentarán los resultados en base a un promediado estadístico de la serie de cinco medidas, obteniendo de este modo unos valores lo suficientemente fiables sobre los que realizar el análisis oportuno.

V. RESULTADOS

Tras realizar las capturas con todas las combinaciones posibles, se procederá al procesado de los ficheros de traza, centrándose el análisis en tres parámetros principales: relación señal a ruido (SNR), errores y retardo (este último está íntimamente relacionado con el rendimiento).

- Relación señal a ruido (SNR)

Se trata de un valor que de alguna manera representa la "calidad" de la señal que llega al dispositivo receptor, empeorando a medida que la distancia del enlace se incrementa. Está ligado a la probabilidad de que se produzca un error en la transmisión de una trama, pudiendo comprobarse que, a medida que la SNR disminuye, la tasa de error aumenta, hasta un valor umbral, por debajo del cual todas las tramas recibidas son incorrectas (experimentalmente se corresponde con unos 15 dB para 802.11g). Los autores en [8] llevan a cabo un estudio experimental en el que se relaciona la SNR con el throughput y con el jitter.

Por motivos de espacio, se ha decidido no incluir las gráficas que muestran la fdp de la relación señal a ruido, ya que la información que se puede obtener de ellas es poco relevante, pues en cualquier caso, para más de un 90% de las tramas su valor es superior a 60 dB, y en ningún momento se sitúa por debajo de los 50 dB.

- Errores

Para una transferencia es relevante la inmunidad que ofrece el canal frente a posibles interferencias, colisiones, etcétera. Debido a los altos valores de la SNR, la FER (tasa

de error de trama) es independiente de la relación señal a ruido, ya que, al ser tan elevada, no es un factor determinante. Se puede asumir, por tanto, que todos los errores son debidos a las colisiones producidas por estaciones o puntos de acceso que utilizan canales adyacentes.

Los datos a tener en cuenta son la FER mencionada anteriormente, que se obtiene como el cociente de las tramas erróneas entre el número total de tramas recibidas, y los valores estadísticos de las longitudes de ráfagas de errores: el máximo, la media y la varianza.

$$FER = \frac{N_{tramas\ erróneas}}{N_{total\ tramas}} \quad (6)$$

- Retardo y throughput

La técnica empleada para la caracterización temporal del canal es la obtención del histograma del retardo [9], comprobando el efecto del mecanismo de acceso al medio (basado en CSMA/CA), pudiendo además extraer valores estadísticos a partir de los cuales es posible estimar otros parámetros como el rendimiento o el jitter de una transmisión.

De manera breve, el método consiste en obtener las diferencias de tiempos entre dos tramas consecutivas, clasificándolas en dos grupos: primeramente, se considerarán tramas correctas cuando el valor del campo "*Identification*" de IP es consecutivo a la última trama recibida, además de no tener error de CRC; por otro lado, se caracterizará el retardo de todas las tramas (correctas + erróneas).

A continuación, se muestran los resultados extraídos para cada una de las extensiones de nivel físico del estándar 802.11 analizadas en este trabajo.

Las unidades por defecto serán microsegundos para los tiempos y megabits por segundo (Mbps) para las velocidades binarias (se trata del rendimiento neto, descontando toda la sobrecarga de la comunicación).

A. 802.11b

Puede apreciarse en los resultados que el comportamiento medido en el laboratorio (Tabla 5 y Figura 5) se encuentra bastante próximo a los valores obtenidos teóricamente, lo cual corrobora la validez de todo el proceso.

Obsérvese que en el proceso de transmisión, la probabilidad de que se produzca un error en una trama es prácticamente nula. Otra característica relevante es que el valor del retardo medio es aproximadamente 100 microsegundos superior al teórico, debido posiblemente a que el escenario de medida no era completamente ideal, existiendo otras estaciones compitiendo por tener acceso al canal. Esto también produce una ligera disminución en el valor del rendimiento final, aún así manteniendo valores muy cercanos al teórico (~ 90%).

El empleo de técnicas adicionales al estándar no sólo mantiene su robustez, sino que origina mejoras importantes en el rendimiento neto, pudiendo llegar a transmitir (con *Fast Frames*) hasta unos 2 Mbps por encima del modo puro.

En resumen, en 802.11b, sobre un canal ideal, la comunicación entre estaciones se producirá de una manera fiable (ausente de errores), y con un rendimiento muy cercano al límite teórico.

B. 802.11g

En este caso (Tabla 6 y Figura 6), las transmisiones muestran siempre una tasa de error de trama que ronda un 10%, a lo que hay que sumarle que se produce una pérdida de aproximadamente 500 tramas por captura (lo que supone un 5% adicional). Estas son tramas que las herramientas empleadas no logran capturar adecuadamente.

La caracterización estadística del retardo también se ve distorsionada debido a tramas que deben esperar para transmitir, en algunos casos varios órdenes de magnitud por encima de los tiempos normales, afectando negativamente a la media y la varianza.

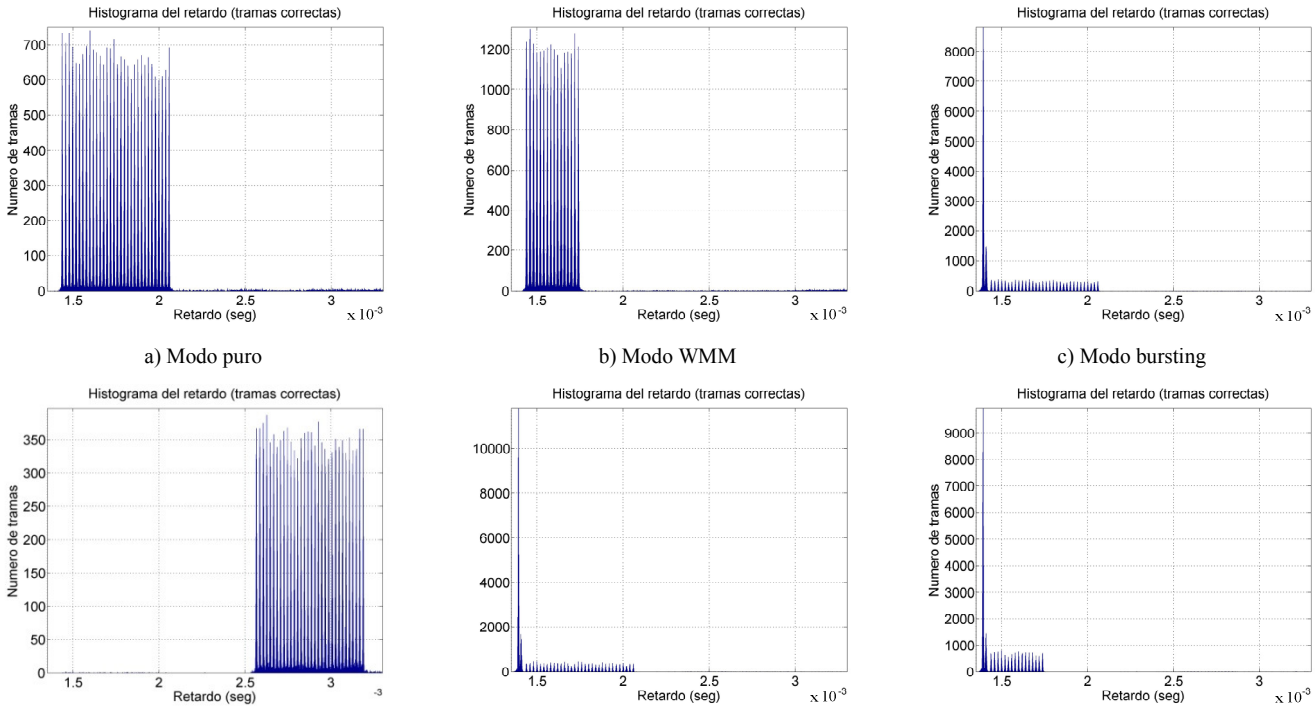
Analizando los histogramas del retardo, el empleo de las técnicas propietarias de Atheros (*Fast Frame y Bursting*) no coincide completamente en su funcionamiento con la caracterización teórica, sobre todo cuando ambas se combinan. Al contrario, se observa que una proporción de las tramas se envía de manera individual, sin emplear el denominado Fast Frame, haciendo más difícil la estimación de su comportamiento. Pueden diferenciarse claramente en las gráficas e) y f) de la Figura 6 los dos comportamientos.

Tabla 5. Resultado promediado sobre 5 medidas de 25000 datagramas UDP cada una (802.11b)

	Puro	*WMM*	*Bursting*	*Fast Frames*	*FF + Bursting*	*Todo*
Tramas capturadas	24996	24996,4	24998,2	12506,4	24996,2	24997,4
Tramas erróneas	1	0,4	0,4	0,2	1,6	1,8
FER	4,001E-05	1,600E-05	1,600E-05	1,599E-05	6,401E-05	7,201E-05
Retardo medio [µseg]	1880,36	1691,54	1636,30	1498,94	1636,30	1567,37
Retardo medio (teór.) [µseg]	1735	1575	1560	2838	1560	1480
Retardo máximo [µseg]	18933,6	14100,8	12043	15952	18595	19341,8
Var. Retardo [μseg^2]	421276,15	259462,05	216391,36	2401799,58	270618,75	235511,05
Throughput [Mbps]	6,26	6,96	7,19	7,85	7,29	7,51
Throughput (teór.) [Mbps]	6,78	7,48	7,54	8,29	7,54	7,95

Tabla 6. Resultado promediado sobre 5 medidas de 25000 datagramas UDP cada una (802.11g)

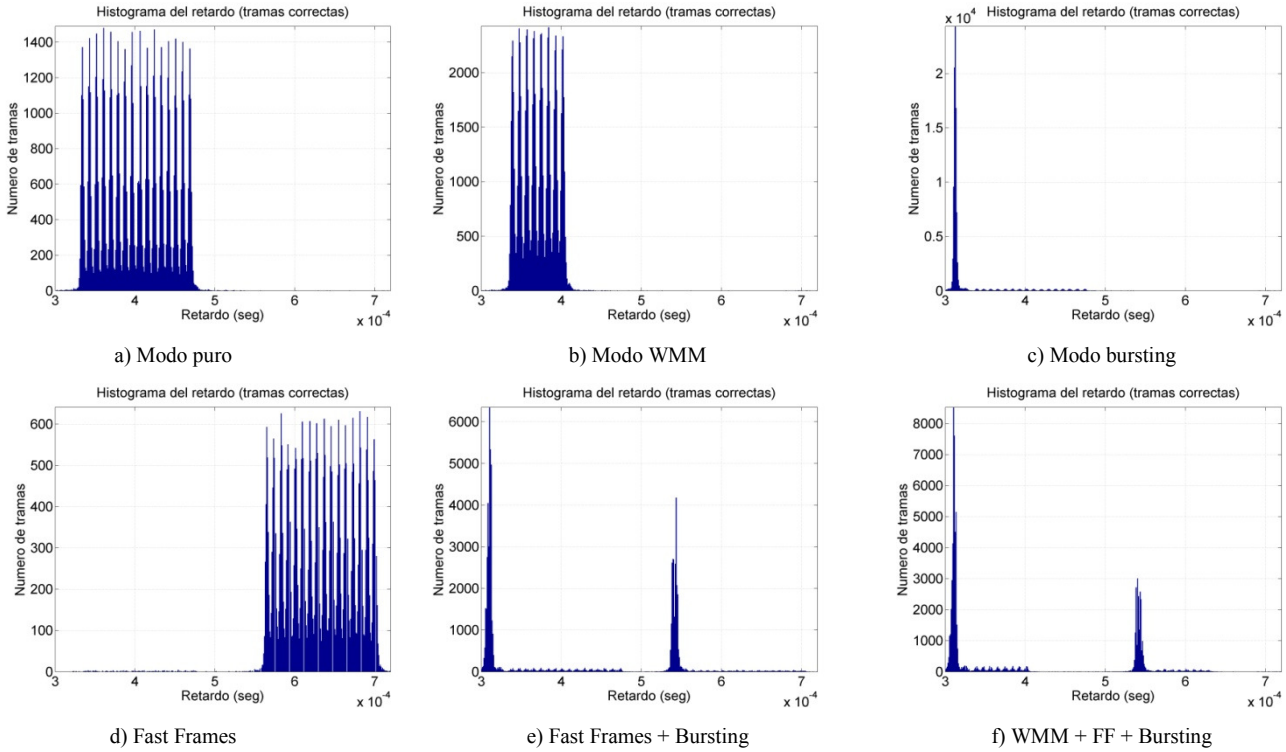
	Puro	*WMM*	*Bursting*	*Fast Frames*	*FF + Bursting*	*Todo*
Tramas capturadas	24550,8	24478,6	24746	12297,6	17532,6	18616,8
Tramas erróneas	2645,4	2615	1920,6	1775,8	1775,6	1670,8
FER	0,1078	0,1068	0,0776	0,1444	0,1015	0,0897
Retardo medio [µseg]	513,748	468,64	387,6	401,74	361,76	370,80
Retardo medio (teór.) [µseg]	390,16	354,16	316	607	543,72	534,72
Retardo máximo [µseg]	379669,4	271058	276745	17648,2	194331,4	433410,8
Var. Retardo [μseg^2]	40383762,37	17621122,14	19067558	311926,56	9335923,48	24156994
Throughput [Mbps]	23,932	25,87	31,53	29,314	33,49	33,76
Throughput (teór.) [Mbps]	30,18	33,25	37,26	38,74	43,31	44,04



a) Modo puro b) Modo WMM c) Modo bursting

d) Fast Frames e) Fast Frames + Bursting f) WMM + FF + Bursting

Fig.5. Histograma del retardo 802.11b



a) Modo puro b) Modo WMM c) Modo bursting

d) Fast Frames e) Fast Frames + Bursting f) WMM + FF + Bursting

Fig. 6. Histograma del retardo 802.11g

Además, donde se utiliza Fast Frames, se puede comprobar que el retardo medio es muy inferior al teórico; se explica porque Wireshark captura dos tramas en lugar de una, con 1 microsegundo de diferencia entre ambas, con lo que altera los resultados.

VI. Conclusiones y Líneas Futuras

La gran eclosión que se ha producido en el uso de la tecnología de comunicaciones inalámbricas IEEE 802.11 ha tenido, entre otras, dos consecuencias principales: por un lado la gran proliferación en el despliegue de este tipo de redes y, por otro, la mejora paulatina de los protocolos definidos en la norma de referencia, con el objetivo de mejorar sus prestaciones, adaptándolas tanto a los protocolos de capas superiores como a los requerimientos de nuevos servicio que demandan cada vez prestaciones más exigentes.

Es por ello que el tipo de análisis que se ha llevado a cabo en este trabajo es fundamental, pues permite analizar de manera más realista cuál es el comportamiento de esta tecnología, siendo especialmente relevante el carácter empírico que se ha seguido. Se ha realizado un análisis exhaustivo del incremento de las prestaciones que tres nuevos procedimientos de capa MAC pueden aportar sobre las recomendaciones de capa física más empleadas en la actualidad: 802.11b y 802.11g y que, además, formarán parte de la futura ampliación 802.11n.

Primeramente se ha realizado un estudio teórico, que ha servido para corroborar la validez del proceso de medidas abordado con posterioridad. Así, y aunque a priori las prestaciones de 802.11g deberían ser claramente superiores a las de 802.11b (al menos en el rendimiento), se ha comprobado que, desde un punto de vista experimental, presenta unas desventajas claras en términos de robustez, lo que podría limitar su uso (al menos a la tasa binaria máxima de trabajo definida en el estándar). Esta falta de fiabilidad, que se agrava por la presencia de otras redes en el entorno de medida (aspecto este que es imposible eliminar en la actualidad en cualquier escenario plausible), hace que se produzcan notables diferencias frente a los valores obtenidos de manera teórica.

Lo que sí que se comprueba es que la aplicación de las técnicas de capa MAC que se han analizado puede aportar un beneficio considerable, en términos del rendimiento que es posible alcanzar, independientemente de la recomendación de capa física empleada.

Una vez que se dispone de un conocimiento exhaustivo del comportamiento en un escenario cercano al ideal, sería conveniente afrontar un estudio más pormenorizado en un entorno más real, donde la distancia penalice la calidad de la transmisión, haya obstáculos intermedios, interferencias, y múltiples dispositivos y redes contendiendo por el uso del canal. En trabajos futuros se ampliará el banco de medidas utilizado en este artículo, para comprobar cuál es el efecto de esos factores en el comportamiento de las redes IEEE 802.11.

Una de las posibles mejoras a introducir en la plataforma de medidas, aprovechando las modificaciones que ya se han llevado a cabo en el driver, vendría de la obtención directa de las estadísticas en el propio driver, sin que sea necesario disponer de ninguna herramienta adicional

Agradecimientos

Los autores desean expresar su agradecimiento al Proyecto Nacional de I+D del Ministerio de Educación y Ciencia titulado "Optimización de Técnicas de Descubrimiento de Servicios sobre Plataformas Inalámbricas Heterogéneas" (TEC2006-05819).

Referencias

[1] Y. Haddad, G. Le Grand, "Throughput analysis of the IEEE 802.11e EDCA on a noisy channel in unsaturated mode", 3rd ACM Workshop on Wireless Multimedia Networking and Performance Modeling , October 22 - 22, 2007.

[2] H. Lee; I. Tinnirello; J. Yu; S. Choi, "Throughput and delay analysis of IEEE 802.1le Block ACK with channel errors" 2nd International Conference on Communication Systems Software and Middleware, COMSWARE 2007, Jan. 2007

[3] T. Li, Q. Ni, T. Turletti, Y. Xiao, "Performance analysis of the IEEE 802.11e Block ACK scheme in a noisy channel," 2nd International Conference on Broadband Networks, BroadNets 2005, pp. 511-517 Vol. 1, Oct. 2005

[4] Y. Do; S. Lee; S. Park, "Adaptive Acknowledgment schemes of the IEEE 802.11e EDCA," 9th International Conference on Advanced Communication Technology, vol.3, pp.1679-1683, Feb. 2007.

[5] IEEE Std. 802.11e, 2005 edition, IEEE Standards for Local and Metropolitan Area Networks, Part 11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Quality of Service Enhancements", Nov. 2005

[6] IEEE Std 802.11n, Draft 9.0, IEEE Standards for Local and Metropolitan Area Networks, Part 11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Enhancements for higher thoughput", Mar. 2009

[7] B. Ginzburg, A. Kesselman, "Performance Analysis of A-MPDU and A-MSDU Aggregation in IEE 802.11n", IEEE 2007 Sarnoff Symposium, Mayo 2007

[8] J. De Bruyne; W. Joseph; L. Verloock; L. Martens, "Evaluation of Link Performance of an Indoor 802.11g Network," *5th IEEE Consumer Communications and Networking Conference, CCNC 2008,* pp.425-429, Jan. 2008

[9] Gilles Berger-Sabbatel, Yan Grunenberg, Martin Heusse, Franck Rousseau, Andrzej Duda, "Interarrival Histograms: A Method for Measuring Transmission Delays in 802.11 WLANs", Research Report, LIG Lab, Grenoble, France, 2007

Defensas frente a ataques DoS a baja tasa contra servidores basadas en políticas de gestión de colas

Rafael Alejandro Rodríguez-Gómez, Gabriel Maciá-Fernández, Pedro García-Teodoro, Jesús Esteban Díaz-Verdejo.
Departamento de Teoría de la Señal Telemática y Comunicaciones, CITIC-UGR,
Universidad Granada,
Calle Periodista Daniel Saucedo Aranda s/n CP 18071 (Granada-España)
rodgom@correo.ugr.es, gmacia@ugr.es, pgteodor@ugr.es, jedv@ugr.es

***Resumen*- En este artículo se evalúa el uso de defensas contra ataques de denegación de servicio a baja tasa contra servidores basadas en políticas de gestión de colas y la viabilidad de su implementación. En un sistema real, para este último fin, se modifica el núcleo del sistema operativo Linux proporcionando un marco de trabajo que permite, de forma flexible y simplificada, la introducción del código que implementa las políticas citadas.**

Se propone una política de gestión de colas y se muestra que su implementación es factible en este núcleo modificado.

Por último, se realizan una serie de pruebas con dicha política de gestión de colas y, a la luz de los resultados obtenidos, se comprueba que es eficaz frente a ataques DoS contra servidores, ya que mitiga sus efectos de forma considerable.

***Palabras Clave*- seguridad, denegación de servicio, gestión de colas, conexión TCP.**

I. INTRODUCCIÓN

Hoy en día se llevan a cabo multitud de operaciones a través de Internet. Muchas de ellas implican tránsito de dinero o de información crucial, y esto conlleva la necesidad de garantizar ciertos niveles de seguridad para que los clientes realicen dichas operaciones. Pero, de la misma forma que existe esta seguridad, existen también una serie ataques que pretenden vulnerarla. La mayoría de éstos tratan de penetrar en un sistema y, de este modo, poder acceder a información privada, e.g., contraseñas de correo electrónico, información de una investigación aún no patentada, números de cuentas bancarias, etc.

Entre los diferentes tipos de ataques existentes, el de *denegación de servicio* (en inglés Denegation of Service o DoS) pretende evitar el acceso a un servicio ofertado de modo total o parcial, mediante el envío de ciertos mensajes hacia uno de los participantes en la comunicación o el propio canal. La mayoría de estos ataques se basan en el envío masivo de peticiones a un participante de la comunicación consiguiendo de esta forma saturarlo, pero se han desarrollado ataques DoS más evolucionados que consiguen su objetivo mediante el envío de peticiones a una tasa de tráfico reducida. El primero en ser desarrollado fue el ataque *Shrew*, diseñado por A. Kuzmanovic [1]. También se pueden citar el ataque de *Reduction of Quality* (*RoQ*) descrito por Guirguis en [2][3], el que M.C. Chan desarrolló para aprovechar las actualizaciones automáticas (descrito en [4]) y G. Maciá-Fernández describió en [5][6] el ataque *DoS a baja tasa contra servidores* (en inglés *Low-Rate DoS Attack against Server, LoRDAS*), que tiene como objetivo reducir la disponibilidad del servidor pudiendo dejar prácticamente sin servicio a todos sus clientes.

En este trabajo se explora un tipo de medidas de defensa frente a este último grupo de ataques (ataques LoRDAS), basada en la aplicación de *políticas de gestión de colas*, entendiéndose éstas como el procedimiento mediante el cual se otorga cierta "inteligencia" a la gestión de las colas del servidor. Se pretende dificultar la tarea parte del atacante de conseguir ventaja sobre los clientes legítimos y así mitigar los efectos del ataque LoRDAS.

En primer lugar, se comprueba si realmente este tipo de medidas de defensa son efectivas contra los efectos dañinos del ataque LoRDAS. Para esto, se diseña una política de gestión de colas sencilla, con el fin de evaluar si ésta es capaz de mitigar los efectos del ataque.

En segundo lugar, se comprueba si las políticas de gestión de colas son una solución factible en cuanto a su implementación en un sistema real. Para este fin, se desarrolla un entorno capaz de incluir en el sistema operativo Linux, de forma flexible y sencilla, cualquier otra política de gestión de colas desarrollada.

Para exponer estas ideas, se divide el artículo en las siguientes secciones. En la Sección II se presentan los fundamentos del ataque DoS contra servidores. En la Sección III se expone la solución propuesta junto con el desarrollo de una política de gestión de colas sencilla. Posteriormente, en la Sección IV, se presenta la implementación de un entorno para inclusión de políticas de gestión de colas en un sistema real. A continuación, en la Sección V, se exponen los resultados derivados de la evaluación de una sencilla política de gestión de colas con objeto de demostrar la eficacia de este tipo de medidas. Por último, en la Sección VI se presentan las conclusiones y unas posibles líneas de trabajo futuro basadas en este trabajo.

II. MODELADO DEL ATAQUE LORDAS

El ataque DoS, en una red de comunicación, tiene como objetivo eliminar o reducir la disponibilidad de un determinado activo mediante la ejecución de determinadas

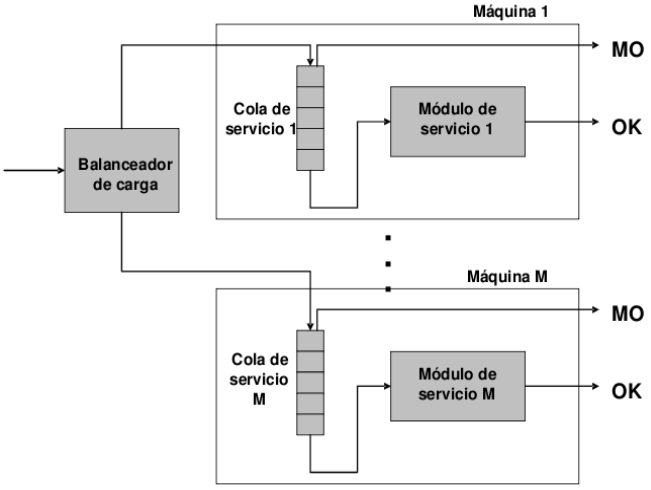


Fig. 1: Modelo de servidor (tomado de [5]).

acciones maliciosas dirigidas a la fuente de información, al canal de comunicación o a ambos. Este tipo de ataques impide o inhibe el uso normal o la gestión de recursos de comunicaciones.

Existen dos métodos fundamentales para la realización de un ataque DoS: la explotación de una *vulnerabilidad* mediante el envío de uno o varios mensajes con una construcción determinada, de manera que sean capaces de aprovecharse de una debilidad existente en el equipo previamente conocida y estudiada y la *inundación* con mensajes de apariencia legítima que acaba por consumir determinados recursos críticos para el funcionamiento correcto del sistema (p.e. tiempo CPU, ancho de banda de red, etc).

Inicialmente los ataques DoS más extendidos eran los de vulnerabilidad. La utilización de estos, con el paso del tiempo, ha ido decreciendo dando paso a un aumento del segundo grupo, los ataques DoS de inundación. Típicamente, para producir la inundación, estos ataques generan una alta tasa de tráfico, lo que los hace detectables por parte de ciertos sistemas de seguridad que basan su detección en un tráfico anómalo de la red. Para evitar la detección por parte de estos sistemas de seguridad, además de para poder atacar sin necesidad de unos recursos abundantes, aparecen los ataques DoS a baja tasa. Este tipo de ataques DoS utiliza la técnica de inundación, pero con la gran diferencia de que no necesitan una alta tasa de tráfico para saturar a la víctima del ataque. Precisamente es esta baja tasa de tráfico la que dificulta la detección de este tipo de ataques.

Para estudiar el ataque contra servidores se define un modelo de servidor consistente en un módulo que recibe mensajes o peticiones, los procesa y emite las respuestas que sean necesarias. Los mensajes recibidos en este servidor pueden proceder bien de los usuarios legítimos, bien de los usuarios malintencionados. El servidor tiene dos tipos de respuestas: los eventos OK y los eventos MO.

- Los eventos OK representan la respuesta que da el servidor a una petición determinada de un cliente en un estado del sistema normal.
- Los eventos MO *mensaje de desbordamiento* (de las siglas en inglés: *Message Overflow*) son la respuesta que da el servidor a los clientes en un estado de sobrecarga de las colas internas o de la memoria de la aplicación.

El modelo completo se ilustra en la Fig. 1 y se ha dividido en las siguientes secciones: un balanceador de carga y M máquinas servidoras, cada una de ellas con una cola de servicio y un módulo de servicio.

Se puede resumir el funcionamiento del modelo siguiendo el camino de una conexión entrante al mismo.

1. La petición llega al balanceador de carga, que elige, según su política interna, la máquina de entre las M posibles a la que enviará dicha petición, exceptuando aquellas máquinas cuya cola se encuentre ocupada completamente. En el caso de que todas las colas se hallen en esta situación el balanceador decidirá aleatoriamente cualquier máquina, ya que el resultado será igualmente la devolución de un evento MO (de desbordamiento).
2. La petición es encolada en la cola de servicio de la máquina elegida por el balanceador de carga siempre y cuando quede alguna posición libre. En caso contrario la petición se descarta y se envía un evento MO.
3. Transcurrido el tiempo necesario en la cola, la petición pasa al módulo de servicio en el que será procesada y tras lo cual se responderá con un evento OK al cliente que inició la petición.

En el ataque LoRDAS, el objetivo básico consiste en evitar la disponibilidad del servidor, haciendo que el tiempo libre de las colas de servicio tienda a cero. Para ello, el atacante debe ser capaz de predecir la aparición de un espacio libre y de insertar una nueva petición suya en cuanto dicha posición aparezca. De este modo, la probabilidad de que los clientes realicen una conexión en ese breve espacio de tiempo es mínima. El atacante estará consiguiendo así que el servidor únicamente sirva sus peticiones y logrará, como consecuencia de esto, denegar el servicio a todos los demás clientes.

En esta situación, el servidor continúa trabajando con total normalidad sirviendo los mensajes que tiene encolados, con la única salvedad de que todos los mensajes que sirve son del mismo cliente, el atacante. Este funcionamiento del servidor, en régimen de normalidad, hace aún más compleja la detección de este tipo de ataques.

El proceso que sigue el atacante para realizar un ataque DoS a baja tasa contra servidores puede resumirse en los siguientes dos pasos:

1. *Saturación* de las colas de servicio. Este paso consiste en llenar todas las colas de servicio de peticiones del atacante.
2. *Captura* del máximo número de posiciones posibles. La captura es el acto de introducir una petición en el momento en el que se libera una posición de alguna cola de servicio, ocupando la posición liberada.

Los procesos de captura y saturación, detallados en [5][6], pueden entenderse como un ataque de vulnerabilidad, aunque el ataque en conjunto es un ataque de inundación al saturar la cola de servicio de la aplicación. El atacante se aprovecha del conocimiento de que en la cola de servicio se libera una posición siempre y cuando el módulo de servicio termine de procesar una petición y envíe el mensaje de respuesta al cliente correspondiente, de modo que, justo tras esta acción, el módulo de servicio que ha respondido,

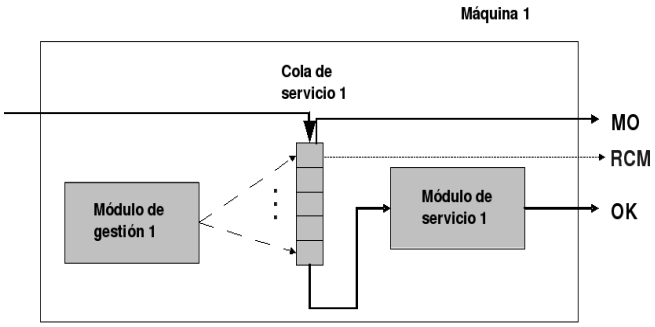


Fig. 2: Modelo de una máquina del servidor con módulo de gestión.

extraerá la siguiente petición de su cola de servicio y generará una posición libre en ella. Esta es la vulnerabilidad que aprovecha el atacante para realizar con éxito el ataque DoS a baja tasa contra servidores.

III. Efecto de las políticas de gestión de colas

La idea presentada en este artículo con objeto de mitigar los efectos del ataque DoS contra servidores a baja tasa se basa en eliminar la ventaja que tiene el atacante sobre los clientes legítimos. Esto implica eliminar el conocimiento que posee el atacante acerca del momento en el que se generará una posición libre en la cola de servicio. Para esto, se propone la inclusión de una inteligencia en la gestión de la cola de servicio, una llamada *política de gestión de colas*.

Denominamos política de gestión de colas a aquel mecanismo que realiza un manejo de las colas de servicio con un fin concreto. En nuestro caso, descartar o priorizar peticiones de la cola de servicio ante la aparición de determinados estados en la misma. La elección de las peticiones que son descartadas o priorizadas sigue, también, una estrategia determinada. Esta actuación pretende conseguir ecuanimidad entre los usuarios, evitando, por tanto, que los clientes maliciosos capturen mayor número de posiciones en la cola que los clientes legítimos.

Para incluir la funcionalidad de las políticas de gestión de colas, al modelo de servidor presentado anteriormente en la Fig. 1 se añade un módulo denominado *módulo de gestión*, que será el encargado de aplicar la política de gestión de colas (Fig. 2). Por tanto, se añade también un evento más a las respuestas del servidor, el *evento RCM solicitud de modificación de conexión* (de las siglas en inglés *Request Connection Modification*). Este evento se produce en el servidor cuando el módulo de gestión determina que una conexión de las encoladas debe ser priorizada o descartada de la cola de servicio. Éste también podría dar como resultado el envío, al cliente correspondiente, de un mensaje *RCM* indicando que su petición será descartada o priorizada. Éste envío es opcional y dependerá de la política de gestión de llevarlo a cabo o no.

En resumen, el funcionamiento del módulo de gestión consiste en monitorizar la cola de servicio y, tras la recogida de cierta información de interés sobre el estado de la cola, procesarla y actuar de acuerdo a su política interna. La política interna que rige el módulo de gestión es la clave del éxito de esta solución. Una buena política de gestión de colas será aquella que extraiga el mayor número de peticiones malintencionadas del atacante, descartando, además, el menor número de peticiones de los clientes legítimos. Si se consigue que los clientes puedan seguir realizando conexiones con normalidad se habrán logrado mitigar los efectos del ataque DoS.

A. Diseño de una política de gestión de colas

A continuación se propone una política de gestión de colas desarrollada para comprobar la efectividad de este tipo de medidas frente a los ataques LoRDAS. El proceso seguido por la política propuesta se puede ver gráficamente en la Fig. 3, en la que se presenta el diagrama de flujo de la misma.

La notación seguida es la siguiente:

- `n_act`: número de peticiones encoladas en la cola de servicio.
- `n_max`: número de posiciones totales de la cola de servicio.
- `n_umb`: umbral de la cola de servicio a partir del cual se aplicará una determinada estrategia de descarte.
- `N`: número de posiciones de la cola que serán descartadas (`N<n_act`).

La política desarrollada se denomina *Descarte Aleatorio de Conexiones* (en inglés *Random Throw Connections RTC*). Se basa en monitorizar la ocupación de la cola hasta que ésta supere el umbral establecido `n_umb`. Entonces, cuando `n_act>n_umb`, se eligen `N` de las conexiones encoladas al azar y se descartan. El umbral se especifica como una proporción del tamaño máximo de la cola.

Se presenta a continuación el pseudo-código de la política RTC:

```
repetir siempre{
   si n_act es mayor que n_umb{
      elegir al azar N peticiones de n_act;
      descartarlas;
   }
}
```

Como se puede comprobar, esta política es una estrategia muy sencilla que no ha sido diseñada para optimizar su rendimiento, sino que se presenta con el único objetivo de evaluar si este tipo de medidas es eficaz frente al ataque LoRDAS. Por tanto, la exploración exhaustiva de las posibles medidas de este tipo no es el objetivo de este trabajo.

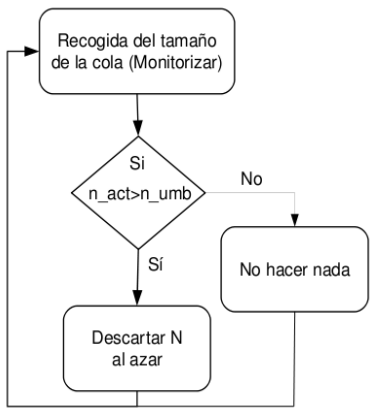


Fig. 3: Diagrama de flujo de la política de gestión de colas RTC.

Como ejemplo de lo anteriormente dicho, nótese que, en la estrategia propuesta, la monitorización del estado de la cola se basa exclusivamente en la observación del número de posiciones ocupadas en cada instante. Sin embargo, podrían existir otras alternativas como, por ejemplo, monitorizar el número de posiciones de un mismo usuario, el número de posiciones de diferentes usuarios dentro de una misma subred, etc.

B. Mitigación de los efectos DoS

Ahora se aborda la cuestión de cómo influye la política RTC en la mitigación del efecto dañino del ataque LoRDAS, es decir, cómo tras la aplicación de la política de gestión de colas, un cliente legítimo puede acceder a los recursos del servidor con una mayor disponibilidad que en el caso de que esta política no fuera aplicada.

Como se ha visto anteriormente el ataque LoRDAS se basa en dos procesos principalmente:

- Mantener las colas siempre *saturadas*, sin posibilidad de que nuevos clientes puedan hacer conexiones.
- *Capturar* todas las posiciones que se vayan habilitando en la cola de servicio prediciendo los instantes en que éstas se generan.

La política RTC dificulta ambos procesos. La saturación de las colas se hace más compleja para el atacante, ya que la política determina un tamaño umbral a partir del cual, cuando el número de peticiones en cola lo supera, un cierto número de peticiones de las encoladas son expulsadas. De esta manera, se dificulta la tarea de mantener las colas siempre saturadas, ya que el servidor libera posiciones siempre que se supere el umbral y estas posiciones pueden ser ocupadas tanto por el atacante como por cualquier cliente legítimo.

Adicionalmente, esta política dificulta la captura de todas las posiciones de la cola por parte del atacante ya que, aunque éste pueda predecir los instantes en que el servidor envía respuestas a los clientes y, por tanto, las posiciones de la cola liberadas de una manera normal por el servidor, no sucede lo mismo para las peticiones que son expulsadas por la política de gestión de colas, ya que esta expulsión genera `N` posiciones libres. Por tanto, con la aplicación de esta política, se consigue que la vulnerabilidad que supone conocer la aparición de una posición libre en la cola de servicio tras cada petición servida por el servidor disminuya de importancia, ya que ahora, no es éste el único modo en el que aparecen posiciones libres.

Esta política de gestión de colas, aunque sencilla, es capaz de hacer frente a complejos mecanismos utilizados por los atacantes para predecir el momento en el que una posición será liberada de la cola. En los resultados mostrados en la Sección V se expondrá en qué medida es capaz de reducir realmente los efectos del ataque LoRDAS.

IV. ENTORNO PARA LA INCLUSIÓN DE POLÍTICAS DE GESTIÓN DE COLAS

Para evaluar la viabilidad de la implementación de la política RTC o cualquier otra política de gestión de colas en un sistema real se diseña y desarrolla un entorno que permite la inclusión de éstas. En este trabajo nos centramos en Linux por ser éste un sistema operativo de código abierto, lo que permite modificar su núcleo adaptándolo a nuestras necesidades.

En esta implementación se supondrá que las peticiones del ataque LoRDAS corresponden a simples peticiones de conexión TCP SYN a un puerto de escucha de la aplicación, tal y como se sugiere en [5]. Por ello, previamente a la implementación de la solución, se estudia cómo se realiza en el núcleo de este sistema operativo el establecimiento de conexión del protocolo TCP/IP.

A. Cola de servicio para TCP en el núcleo 2.6 de Linux

Utilizamos el núcleo en su versión 2.6.24 por ser el más actual en el momento de inicio de este trabajo. Para este núcleo la implementación de la cola de servicio para conexiones TCP se divide en dos colas diferentes [7] (ver Fig. 4):

- *Cola de conexiones incompletas.* Tiene una entrada por cada segmento TCP SYN recibido. Para estas conexiones el servidor se encuentra en el estado SYN_RCVD.
- *Cola de conexiones completadas*. En ésta se encuentra una entrada por cada conexión para la que se ha completado correctamente la negociación en tres pasos, con lo que se encuentra en el estado ESTABLECIDA.

Si bien la cola de conexiones incompletas afecta al comportamiento de la entrada de las conexiones a la aplicación, la implementación que realmente corresponde a la cola de servicio del modelo del servidor es la de conexiones completadas. A esta cola se accede por dos vías diferentes (Fig. 4):

- *Acceso desde la red:* se añaden conexiones a la cola de conexiones completadas al pasar del estado SYN_RCVD al estado ESTABLECIDA. Esto se produce al recibir el ACK como respuesta final de la negociación en tres pasos, siempre y cuando quedase espacio en la cola.
- *Acceso desde la aplicación:* se extraen peticiones por parte de la aplicación por medio de la llamada al sistema *accept*. En este caso, la primera conexión de la cola pasa al espacio de usuario.

La cola de conexiones completadas es una estructura del tipo `request_sock_queue` (ver Fig. 5) que representa una lista doblemente enlazada. Los punteros de la lista apuntan a

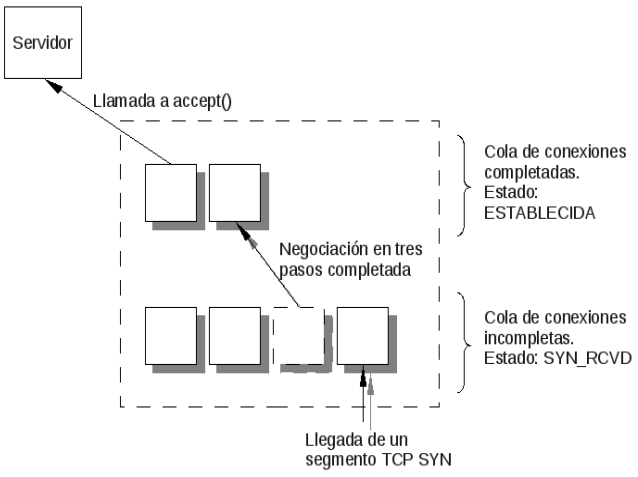


Fig. 4: Colas de conexiones en el núcleo 2.6 de Linux.

estructuras del tipo `request_sock`, que son los elementos que componen la cola y que, por tanto, representan las conexiones establecidas. Éstos contienen básicamente un puntero al siguiente elemento de la cola (`dl_next` en Fig. 5) y un puntero a una estructura `sock` que es la que contiene la información de la conexión.

La sección de código del núcleo en la que se encuentra implementada la fase final de la negociación en tres pasos corresponde a la función `tcp_v4_syn_recv_sock`. Esta función es especialmente adecuada para incluir en ella las políticas implementadas porque es la que se ejecuta tras el término correcto de la negociación en tres pasos, devolviendo el nuevo `socket` que será añadido a la cola de peticiones completadas. Es, por tanto, un lugar estratégico desde el que monitorizar la cola de conexiones establecidas y tomar, dependiendo de la política de gestión de colas que se haya implementado, una decisión u otra.

B. Arquitectura de la implementación.

Decidido el lugar en el que se incluirá la modificación del núcleo, se estudia el sistema operativo Linux y las alternativas que existen para modificar aquél.

El núcleo del citado sistema operativo está organizado siguiendo una arquitectura monolítica, en la cual todas sus partes (sistemas de ficheros, manejadores de dispositivos, protocolos de red, etc.) están enlazadas como una sola imagen que es la que se carga y ejecuta en el arranque del sistema.

Esta estructura lo hace bastante eficiente, ya que disminuye los cambios de contexto, es decir, los cambios de espacio de direcciones de usuario a núcleo, para poder ejecutar instrucciones en modo privilegiado. Por contra, ser un sistema monolítico implica también que existe una menor fiabilidad. A modo de ejemplo, la gestión de ficheros del sistema puede tener acceso a la gestión de la tarjeta de red y esto podría ser nefasto e incluso constituir una vulnerabilidad a explotar desde el punto de vista de un posible atacante. También podría dar lugar a un sistema poco flexible ya que, cualquier funcionalidad que se le quisiera añadir al núcleo del sistema, requeriría una recompilación completa del mismo. Para mitigar en gran medida este problema, en la versión 1.2 de Linux, se incluyó el soporte para la carga dinámica de módulos en el núcleo (en inglés *Loadable Kernel Module LKM*). Este nuevo avance permitió la incorporación de una modificación al núcleo sin la necesidad de reiniciar el sistema completo.

La implementación del entorno para la inclusión de políticas de gestión de colas se desarrolla en un módulo del núcleo debido a la versatilidad que éstos ofrecen. Además, la implementación de este entorno en un módulo del núcleo permite un nivel mayor de abstracción, gracias al cual el usuario no tiene porqué introducir el código del entorno en un archivo del núcleo y dentro de él en una función concreta. Tan sólo es necesario implementar la política de gestión de colas en el lenguaje de programación C en un archivo diferente.

C. Módulos de seguridad de Linux (LSM) y garfios.

El núcleo está preparado para insertar módulos en escenarios concretos como *drivers* para dispositivos, sistemas de ficheros o llamadas al sistema. Realizar modificaciones en otro sentido, como por ejemplo en otro lugar concreto del núcleo, requiere de un análisis exhaustivo del código.

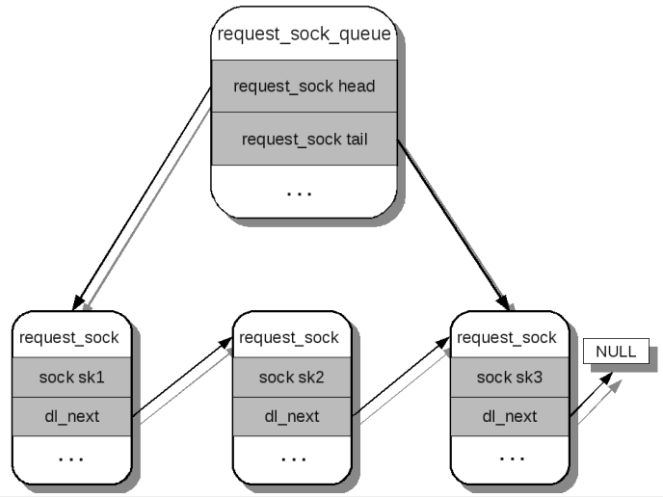


Fig. 5: Estructura de la cola de conexiones establecidas (`request_sock_queue`).

Para incluir una política de gestión de colas se propone modificar el núcleo en la función `tcp_v4_syn_recv_sock`. Sin embargo, modificar el núcleo implicaría un núcleo diferente para cada política de gestión de colas desarrollada, con lo que la solución escogida es una adaptación de un modo concreto de modificación mediante módulos descrito en [8]. Esta modificación es posible gracias al concepto de los garfios (*hooks*), utilizado en los módulos de seguridad de Linux o LSM (del inglés: *Linux Security Modules*).

En el trabajo citado [8] se hace uso de una llamada al sistema creada para este fin y de unas bibliotecas que no son necesarias para la inclusión de políticas de gestión de colas. Por este motivo, ha sido necesario adaptar el método de modificación del núcleo utilizado.

El objetivo de LSM se basa en que ciertas políticas de seguridad sean aplicadas en puntos concretos del núcleo en los que se comprueban, en caso de estar cargado el núcleo, una serie de condiciones para permitir el acceso o no de una petición. Este objetivo es fácilmente extrapolable a las necesidades de un entorno que permita la inclusión de políticas de gestión de colas. Entrando en un mayor nivel de detalle se puede resumir que el objetivo de un garfio es el de, en el punto concreto del código en el que se incluya, llamar a la función del módulo que analiza el estado de la información del núcleo y decide qué actuación llevar a cabo. La gran ventaja que brinda la utilización de los garfios frente a la modificación convencional por medio de módulos es que permiten modificar el núcleo en puntos concretos de su código y, como se ha comentado en el inicio de este punto, éste es precisamente el requisito necesario para la implementación del entorno para la inclusión de políticas de gestión de colas.

Una modificación mediante garfios es una solución intermedia entre la modificación directa del núcleo y un módulo del núcleo. Esto es así porque es necesaria una modificación del núcleo inicial en la que se incluye el garfio en el punto en el que se llamará a la política de gestión de colas mientras que toda la implementación relativa a la política en sí se realiza en un módulo del núcleo.

La modificación llevada a cabo en el núcleo consiste en un elemento de programación que realizará la función de garfio: una estructura denominada `access_control` (cuando se hable de una instanciación concreta de dicha estructura se la llamará `access_cont`). El contenido de esta estructura es de un único elemento: un puntero a función. Este puntero a función es el encargado de apuntar a la llamada `función_inútil`, que no aplica ninguna política hasta que el módulo es cargado, momento en el que pasa a apuntar a la `función_útil` descrita en el módulo que implementa la política de gestión de colas (Fig. 6).

Todo módulo del núcleo debe tener, al menos, dos funciones: `init_module` (ejecutada al cargar el módulo) y `cleanup_module` (ejecutada al descargar el módulo). Para que el garfio funcione correctamente estas funciones deben realizar las siguientes acciones:

- `init_module`: en esta función se guarda el valor del puntero a función de la estructura `access_cont` (que apunta a la `función_inútil`) y se apunta a la `función_útil`.
- `cleanup_module`: se apunta de nuevo el puntero a función a la `función_inútil`, utilizando el valor almacenado en la función `init_module`.

Adicionalmente, para la modificación existen unas consideraciones a tener en cuenta:

- Incluir el fichero de definición de la estructura `access_control`.
- Definir una instancia de esta estructura con el mismo nombre que se definió en el núcleo (`access_cont`) para que se reconozca como el mismo objeto.
- Esta instancia debe estar precedida por la palabra clave `extern`, que indicará que este objeto ha sido implementado externamente.

V. EVALUACIÓN DE LA POLÍTICA RTC

Expuestos los conceptos básicos del modelo de servidor y el ataque LoRDAS, explicada la política de gestión de colas implementada, presentadas las estructuras del núcleo de Linux que intervienen en el desarrollo del entorno para la inclusión de políticas de gestión de colas y analizado en sí el entorno y el método utilizado para incluirlo en el núcleo del sistema operativo Linux, se puede abordar la evaluación de la política propuesta.

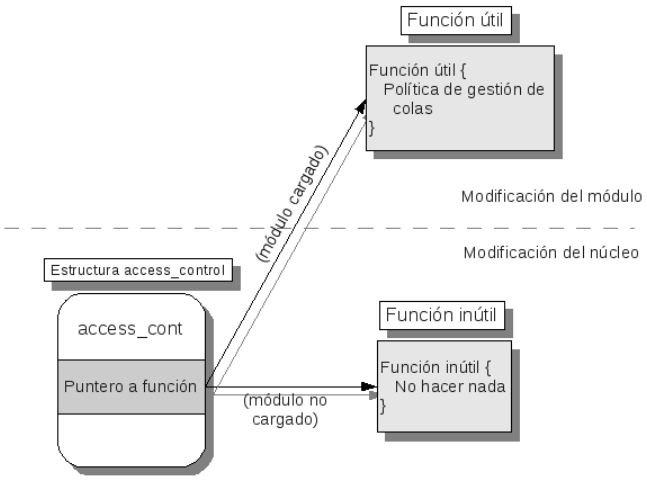


Fig. 6: Funcionamiento de un garfio.

A. Descripción del entorno de pruebas

Para hacer posible la evaluación de la política de gestión de colas y, por ende, del entorno de inclusión de medidas de defensa ante ataques DoS, se implementa un sistema cliente-servidor multihebra. Éste tendrá la finalidad de recoger datos del funcionamiento del sistema frente a un ataque LoRDAS, siendo aplicada la política de gestión de colas como sin serlo.

Se implementa un servidor que acepta conexiones cada 5 segundos. Esto implica que cada 5 segundos, si hay alguna conexión en la cola de conexiones establecidas (o en el modelo general en la cola de servicio), se atenderá la primera conexión de la cola, habilitándose, por tanto, una posición libre en ésta.

Se generan dos programas cliente. Uno de ellos será el atacante y otro el cliente legítimo. Ambos clientes, para poder ser comparados, tendrán la misma tasa de generación de peticiones de conexión, siendo ésta de 1 conexión cada 5 segundos. Esto provocará un desbordamiento de la cola de conexiones establecidas, puesto que a ésta llegará, en media, una petición de conexión cada 2,5 segundos, mientras que la tasa de servicio es de 1 conexión cada 5 segundos. El objetivo es evaluar si ambos clientes tienen las mismas oportunidades de ser servidos y, por tanto, si se encuentran en una situación justa, o si, por el contrario, el atacante es servido con una prioridad superior.

El cliente legítimo es implementado de modo que no genera solicitudes de conexión de una manera determinista, sino que el tiempo entre solicitudes sigue una distribución de de Poisson de media 5 segundos. Para representar la acción del atacante, éste utilizará algún mecanismo que le permita tener ventaja sobre el cliente legítimo y así, dificultar o impedir las conexiones de éste (objetivo del ataque DoS). Concretamente, el cliente malicioso realiza un ataque de tipo LoRDAS como se ha descrito con anterioridad. En nuestro caso la implementación del ataque consiste en enviar una petición de conexión en el momento justo en el que el servidor finaliza el servicio de una petición y, en consecuencia, extrae una nueva petición de la cola de servicio. Como se ha comentado previamente en la Sección II, el conocimiento de la liberación de posiciones de la cola se consigue realmente a través de un complejo mecanismo [5] [6]. En nuestro caso solamente se pretende simular que esto sucede, de modo que en los instantes en los que el servidor acepta una conexión y, por tanto, libera una posición de la cola de conexiones establecidas, nos limitaremos a escribir en un archivo llamado *secreto*. El atacante monitoriza este archivo y, en el instante en el que descubre una modificación en él, realiza un intento de conexión con el servidor. De esta forma, el atacante aumenta la probabilidad de adelantarse al cliente legítimo, capturando la inmensa mayoría de las posiciones de la cola y consiguiendo denegar el servicio.

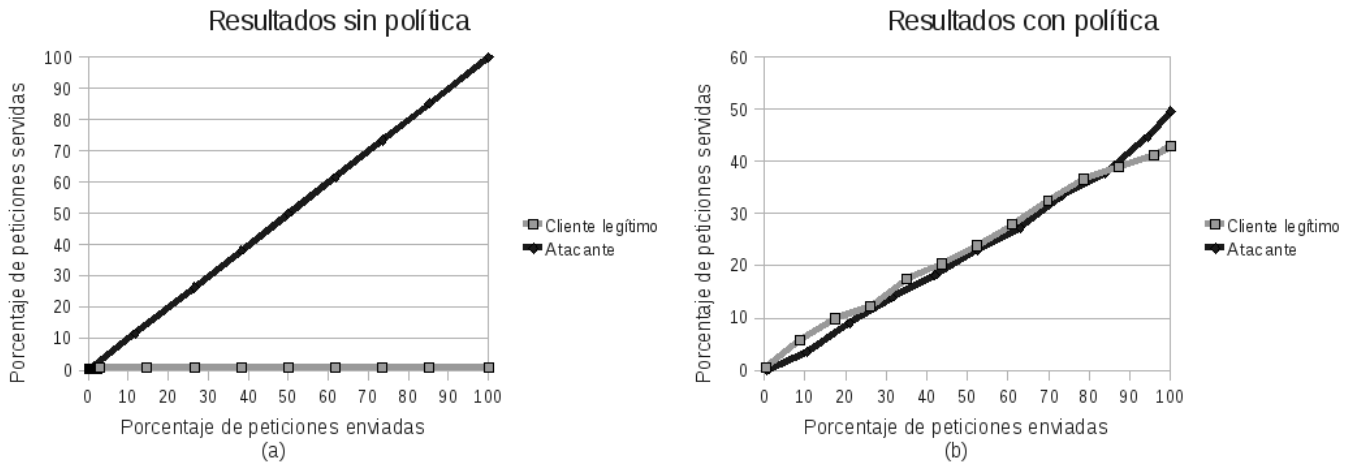


Fig. 7: Porcentaje de conexiones servidas con un tamaño de cola igual a 3: (a) sin aplicar la política y (b) con la política activa.

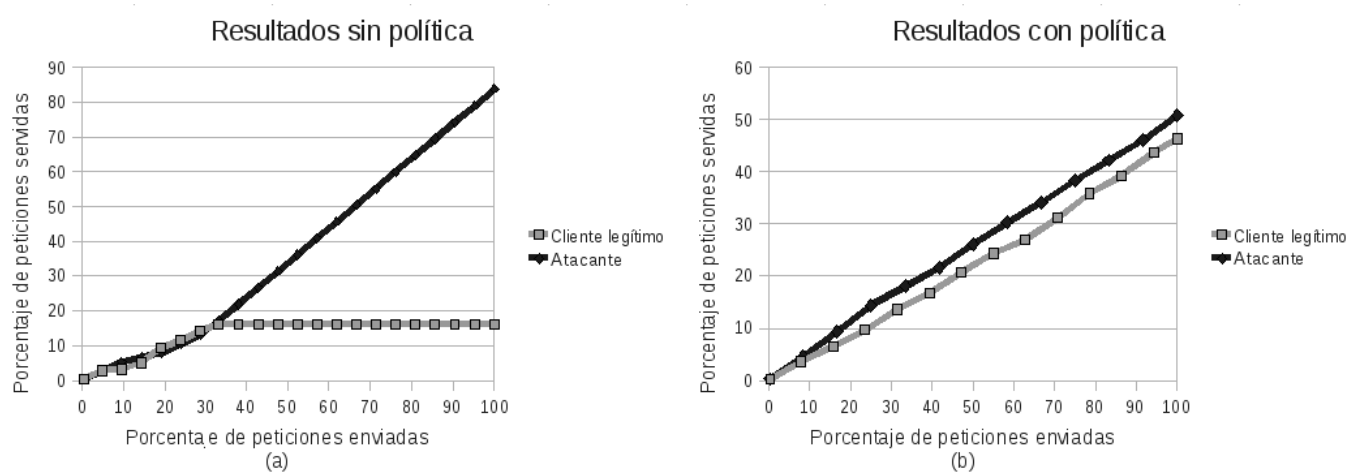


Fig. 8: Porcentaje de conexiones servidas con un tamaño de cola igual a 51: (a) sin aplicar la política y (b) con la política activa.

B. Resultados

Las pruebas realizadas consisten en lanzar la aplicación servidora junto con ambos clientes (legítimo y atacante), con el módulo descargado en primera instancia (política no activa). Tras un tiempo de ejecución suficientemente elevado para que la información recogida sea relevante, se interrumpen todas las aplicaciones y se lanzan de nuevo, pero en esta ocasión se carga el módulo, lo que implica que se aplicará la política de gestión de colas. El entorno implementado imprime en su ejecución el número de peticiones servidas con respecto a los intentos realizados.

Se realizan 2 pruebas: una con tamaño de cola de conexiones establecidas igual a 3 y otra con un valor igual a 51. En la Fig. 7 se ilustran los resultados del tamaño de cola 3. Para tamaño 51 éstos se pueden ver en la Fig. 8.

La notación utilizada puede resumirse en:

- $E(t)$: número de peticiones enviadas desde el inicio de la evaluación hasta el instante t.
- E_T: número de peticiones enviadas en total.
- $S(t)$: número de peticiones servidas desde el inicio de la evaluación hasta el instante t.
- S_T: número de peticiones servidas en total.

En las Fig. 7 y 8, el eje de ordenadas representa el porcentaje de peticiones enviadas hasta el instante *t* respecto del total ($100*E(t)/E_T$) y el eje de abscisas representa el porcentaje de peticiones servidas hasta el instante *t* respecto al total ($100*S(t)/S_T$). Esto quiere decir que todas las representaciones recorrerán todo el eje de ordenadas, ya que van desde un porcentaje de intentos cero al total (100%). Se presentan los resultados en porcentaje precisamente para que el eje de ordenadas sea el mismo tanto para el cliente legítimo como para el atacante. Esto es de utilidad ya que no realiza los mismos envíos de peticiones de servicio un cliente que el otro aún teniendo la misma tasa. En efecto, el cliente legítimo, al generar las conexiones según una distribución de Poisson, en ocasiones realizará más de un intento cada 5 segundos y en otras ocasiones menos, aunque en media el número de intentos será el mismo.

Los resultados obtenidos para la ejecución de las pruebas, ya sea con tamaño de cola igual a 3 o igual a 51, en caso de no aplicarse la política de gestión de colas, demuestran que el atacante consigue realizar con éxito su ataque, ya que el cliente no es capaz de realizar conexiones a partir del instante de saturación de la cola. Esto se debe a que el atacante captura con éxito todas las posiciones que se liberan de la cola de conexiones establecidas y mantiene siempre saturada dicha cola. Para el caso de 51 posiciones de cola, aún sin activar la política de seguridad, sucede lo mismo, pero con la diferencia de que la cola de conexiones establecidas es de un

tamaño mayor, permitiendo de este modo observar el período en el que la cola es saturada (hasta un porcentaje de peticiones enviadas ≈ 30%). A partir de este porcentaje, el atacante consigue capturar todas las posiciones que se liberan de la cola evitando el acceso a ésta del cliente legítimo y denegando, por tanto, su servicio. Este hecho puede verse claramente en las Figs. 7(a) y 8(a) ya que el cliente legítimo, una vez se han saturado las colas, no varía su porcentaje de conexiones servidas.

En el caso de ser cargada la modificación del núcleo y, por consiguiente, de aplicarse la política de gestión de colas RTC, el atacante no obtiene el mismo éxito en su ataque DoS, como puede verse en las Figs. 7(b) y 8(b). De hecho, al término de la ejecución de la aplicación, el porcentaje de peticiones enviadas respecto al de peticiones servidas de ambos clientes ronda el 50 %, lo que implica que la política ha conseguido otorgar las mismas posibilidades de acceso a la cola a los dos clientes y mitigar, por consiguiente, los efectos del ataque LoRDAS.

VI. Conclusiones y trabajo futuro

Como conclusión de este trabajo, podemos resaltar que se ha demostrado la eficacia de la aplicación de políticas de gestión de colas frente a ataques DoS contra servidores a baja tasa. Para evaluar esta eficacia se ha implementado una política de gestión de colas (RTC) que, aunque sencilla, ha sido capaz de mitigar en gran medida los efectos de los ataques LoRDAS. También es reseñable el diseño e implementación de un entorno que hace factible la aplicación de este tipo de medidas de defensa en un sistema real, concretamente el núcleo del sistema operativo Linux.

Como trabajos futuros podemos destacar:

- La generación de políticas de seguridad que introduzcan probabilidades de descarte en casos concretos, que utilicen mecanismos de detección de atacantes para realizar el descarte de la conexión, etc. La evaluación de estas políticas para determinar aquella que dificulte, de la mejor forma posible, los efectos de la realización de un ataque DoS.
- La adaptación del entorno para la inclusión de medidas de defensa frente a ataque DoS basadas en políticas de gestión de colas a otros sistemas operativos de código abierto, como puede ser OpenBSD.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el MICINN mediante el proyecto TEC2008-06663-C03-02.

Referencias

[1] Kuzmanovic, A., Knightly E., Low-rate TCP-targeted denial of service attacks and counter strategies. IEEE/ACM Trans Network 14(4): pp. 683-96, 2006.

[2] Guirguis M, Bestavros A, Matta I, Zhang Y. Reduction of quality (RoQ) attacks on internet end-system, INFOCOM 2005. In: 24th Annual joint conference of the IEEE computer and communications societies; 2005. pp. 1362-72.

[3] Guirguis M, Bestavros A, Matta I, Zhang Y. Reduction of quality (RoQ) attacks on dynamic load balancers: vulnerability assessment and design tradeoffs, INFOCOM 2007. pp. 857-65.

[4] Chan MC, Chang E, Lu L, Ng S. Effect of malicious synchronization, ACNS, Singapore, Jun 6-9, 2006. In: Springer Lecture Notes in Computer Science, vol. 3989; 2006. p. 114-29.

[5] Maciá-Fernández, G., Díaz-Verdejo, J.E., García-Teodoro, P., Evaluation of a Low-Rate DoS Attack Against Application Servers, *Computers & Security*, Vol. 27; pp. 335-354, 2008.

[6] Maciá-Fernández G., Díaz-Verdejo J.E., García-Teodoro P., Evaluation of a low-rate DoS attack against iterative servers. Computer Networks 2007;51(4):1013-30

[7] Stevens, W.R., Fenner, B., and Rudoff, A. M., (2004) UNIX Network Programming, the sockets networking API, Volume 1. Addison-Wesley Professional, 3rd edition, ISBN: 0-13-141155-1.

[8] Wright, C., Cowan, C., Morris, J., Smalley, S., Kroah-Hartman, G., Linux security modules: General security support for the Linux kernel, Proceedings of the 11th USENIX Security Symposium, pp. 17-31, 2002.

Verificación Formal Automatizada del Protocolo de Firma de Contratos FPH Usando Colored Petri Nets

Andreu Pere Isern-Deyà, M. Magdalena Payeras-Capellà, Macià Mut-Puigserver,
Josep Lluís Ferrer-Gomila, Llorenç Huguet-Rotger
Departament de Matemàtiques i Informàtica, Universitat de les Illes Balears
Cra. de Valldemossa, km 7.5. Palma (Illes Balears)
{andreupere.isern, mpayeras, macia.mut, jlferrer, l.huguet}@uib.es

***Resumen*—Un caso particular de los protocolos de intercambio equitativo es la firma de contratos, en la que los participantes se comprometen a firmar un contrato acordado condicionando su firma a que todos obtengan el contrato firmado o ninguno de ellos lo tenga. Un trabajo importante recae en la verificación formal de estos protocolos para probar sus propiedades y la inexistencia de vulnerabilidades. En este artículo presentamos un modelo de verificación formal automatizada, que va a ser aplicado sobre el protocolo FPH para demostrar que mantiene la equidad frente a una serie de ataques préviamente definidos.**

I. Introducción

La firma de contratos es un caso particular de un protocolo de intercambio equitativo, en el que dos o más participantes se comprometen a firmar un contrato préviamente acordado, con la condición que a su finalización, todos tengan el contrato firmado o ninguno de ellos lo tenga. Por tanto, un protocolo equitativo siempre debe proporcionar un trato igualitario a todos los participantes. Además, debe generar evidencias que prueben el comportamiento de los usuarios, para que, en caso de disputa, un árbitro externo pueda evaluar las evidencias y tomar una decisión sin ambigüedades.

Los protocolos de firma de contratos normalmente hacen uso de una Tercera Parte de Confianza (TTP) que ayuda a los usuarios a finalizar satisfactoriamente el intercambio. La TTP puede involucrarse en diferente medida en la ejecución del protocolo, actuando en cada paso, o bien solo interviniendo cuando los usuarios lo requieran. Usando este tipo de TTP, se han definido varios protocolos eficientes y optimistas [1] que requieren solo tres mensajes para finalizar su ejecución. Los protocolos de Micali [2] y el protocolo FPH [3] cumplen estas dos propiedades.

Con el objetivo de verificar formalmente un protocolo de intercambio equitativo, se han ido utilizando técnicas manuales, complicadas de aplicar y que deben ser adaptadas en cada caso, como el uso de *Strand Spaces* [4]. Usando estas técnicas, Bao describe [5] tres ataques al protocolo de Micali y propone un protocolo mejorado. Recientemente, Sornkhom y Permpoontanalarp [6] han aplicado un método automatizado para analizar la seguridad del protocolo de Micali mediante el uso de Coloured Petri Nets (CPN) [7]. El método permite la demostración de las vulnerabilidades del protocolo de Micali descubiertas por Bao, además de detectar dos nuevos ataques a Micali.

En este artículo vamos a aplicar un nuevo modelo de verificación formal y automatizada de protocolos de firma de contratos, basado en [6], que permitirá analizar con mayor rapidez y generalización las propiedades de los protocolos de intercambio equitativo. Como primer paso, mediante la aplicación de dicho modelo, somos capaces de verificar formalmente la conservación de la equidad de FPH frente a los ataques descritos por Bao, Sornkhom y Permpoontanalarp en sus respectivos trabajos.

II. Protocolo de firma FPH

II-A. Propiedades Ideales del Protocolo de Firma de Contratos

Las soluciones prácticas para firma de contratos requieren de la existencia y la posible participación de una TTP. Para que el protocolo sea eficiente, generalmente se desean conseguir tres objetivos:

- Reducir la participación de la TTP.
- Reducir el número de mensajes intercambiados.
- Minimizar el requerimiento de operaciones costosas y de gran volumen de información cuando la TTP deba actuar.

El primer objetivo es alcanzado en varias soluciones optimistas [1], [8], [9], [10], en donde la TTP sólo actua en caso de conflicto entre las partes para garantizar la equidad del protocolo. En referencia al número de mensajes intercambiados entre dos partes se establece que el mínimo son tres para las propuestas optimistas [1], [11].

Además de los anteriores objetivos, un protocolo tiene que proporcionar evidencias a las partes implicadas que prueben, a la finalización del intercambio, si el contrato ha resultado firmado. Otras propiedades adicionales que deberían ser cumplidas por protocolos optimistas són [9], [10]:

- *Efectividad*. Si las partes implicadas actúan correctamente, la TTP no debe de participar.
- *Equidad*. Ningún participante debe obtener una situación de ventaja en ninguna de las etapas de la ejecución del protocolo.
- *Asincronía*. Los participantes pueden decidir cuando finalizar la ejecución del protocolo, sin límites de tiempo.
- *No repudio*. Los participantes no pueden negar sus acciones.

X, Y	Concatenación de dos mensajes X e Y
$H(X)$	Función hash de una vía resistente a colisión del mensaje X
$S_i(X)$	Firma digital del mensaje X con la clave privada de i (aplicando préviamente un hash sobre el mensaje X)
$i \rightarrow j : X$	i envía un mensaje X a j
M	Mensaje conteniendo el contrato que debe ser firmado. Especifica el iniciador, A, y el receptor B
$h_A = S_A(M)$	Firma de A sobre el contrato M
$h_B = S_B(M)$	Firma de B sobre el contrato M
$ACK_A = S_A(h_B)$	Firma de A sobre h_B; reconocimiento que A conoce que el contrato ha sido firmado. Forma parte de la evidencia de B
$ACK_T = S_T(h_B)$	Firma de TTP sobre h_B; es equivalente al reconocimiento que A debería haber enviado
$h_{AT} = S_A[H(M), h_A]$	Evidencia que A ha solicitado la intervención de la TTP
$h_{BT} = S_B[H(M), h_A, h_B]$	Evidencia que B ha solicitado la intervención de la TTP
$h'_B = S_T(h_B)$	Firma de TTP sobre h_B que prueba su intervención

Cuadro I
NOTACIÓN Y ELEMENTOS DEL PROTOCOLO FPH

- *Verificabilidad de la TTP*. Si la TTP actúa incorrectamente, todas las partes afectadas deben de ser capaces de demostrarlo.

II-B. Descripción del Protocolo de Firma de Contratos FPH

El protocolo de firma de contratos FPH [3] asume que el iniciador, *A*(lice), y el receptor, *B*(ob), están deacuerdo en firmar un contrato, *C*, antes de iniciar la ejecución del mismo. El canal de comunicaciones entre las partes firmantes es inseguro, de forma que no se puede asegurar que los mensajes lleguen a su destino sin alteraciones. Por su parte, el canal entre los participantes y la TTP es elástico [12], lo cual significa que los mensajes llegan al destino deseado pero no se puede precedir el tiempo de llegada. La notación y los elementos usados en la descripción del protocolo se especifican en el Cuadro I.

Los participantes, *A* y *B*, intercambiarán las evidencias de no repudio directamente, sin la intervención de la TTP, mediante la ejecución del subprotocolo de *intercambio* (Cuadro II). Sólamente en caso que los participantes no obtengan los elementos esperados de la parte contraria, la TTP será invocada, mediante la iniciación de los subprotocolos de *cancelación* (Cuadro III) y *finalización* (Cuadro IV).

Si la ejecución del protocolo termina, el iniciador *A* habrá obtenido la evidencia NR (no repudio), h_B, y por su parte, *B* tendrá la evidencia NR formada por h_A y ACK_A. Por tanto, el protocolo cumple el requerimiento de efectividad, ya que para completar el protocolo solo son necesarios tres pasos. Si este no es el caso, *A* o *B*, o los dos, necesitan rectificar la situación iniciando el subprotocolo de *cancelación* o de *finalización*, respectivamente, para que de esta forma, la situación se asegure equitativa. Si *A* "afirma" (*A* puede intentar engañar a *B*) que no ha recibido el mensaje 2 de *B*, *A* puede iniciar el subprotocolo de *cancelación* (Cuadro III).

Al inicio del subprotocolo de *cancelación*, la TTP verifica la corrección de la información proporcionada por *A*. Si la información es errónea o incompleta, la TTP envía un mensaje de error a *A*. De otra forma, la TTP procede de una de las dos formas. Si la variable *finished* es *true*, significa que *B* contactó préviamente con la TTP, la cual le contestó con el elemento NR, ACK_T. Por tanto, la TTP recupera el NR recibido de *B*, h_B, y lo envía a A, juntamente con h'_B para demostrar su intervención. Si *B* no hubiera contactado préviamente con la TTP, ésta enviará un mensaje de cancelación a *A*, y guardará esta información (*canceled=true*) para satisfacer posteriores peticiones procedentes de *B*. Sea cual sea el caso, el intercambio llega a una situación equitativa para las dos partes.

Si *B* "afirma" que no ha recibido el mensaje del paso 3, entonces *B* puede iniciar el subprotocolo de *finalización* (Cuadro IV).

En el subprotocolo de *finalización*, de la misma forma que con *A*, la TTP verifica la información proporcionada por *B*. Si determina que es errónea o incompleta, envía un mensaje de error a *B*. De otra forma, procederá de una de las dos maneras posibles. Si la variable *canceled* es *true*, indica que *A* había contactado préviamente con la TTP y por tanto, le había

1. *A* $\rightarrow$ *B*:		M, h_A
2. *B* $\rightarrow$ *A*:		h_B
3. *A* $\rightarrow$ *B*:		ACK_A

Cuadro II
SUBPROTOCOLO DE INTERCAMBIO

	1'. *A* $\rightarrow$ *T*:	$H(M), h_A, h_{AT}$
IF (*finished=true*)	2'. *T*:	extrae h_B almacenado
	3'. *T* $\rightarrow$ *A*:	h_B, h'_B
ELSE	2". *T* $\rightarrow$ *A*:	$S_T("canceled", h_A)$
	3". *T*:	Establece *canceled=true*

Cuadro III
SUBPROTOCOLO DE CANCELACIÓN

	2'. *B* $\rightarrow$ *T*:	$H(M), h_A, h_B, h_{BT}$
IF (*canceled=true*)	3'. *T* $\rightarrow$ *B*:	$S_T("canceled", h_B)$
ELSE	3". *T* $\rightarrow$ *B*:	ACK_T
	4". *T*:	almacena *finished=true* y h_B

Cuadro IV
SUBPROTOCOLO DE FINALIZACIÓN

devuelto un mensaje de cancelación. De forma equivalente, ahora tiene que enviar un mensaje similar a *B*. Si *A* no hubiera contactado anteriormente con la TTP, ésta enviará el elemento NR, ACK_T, a *B*. En este caso, la TTP guardará el NR, h_B, y asignará el valor de la variable *finished* a *true* con el objetivo de satisfacer posteriores peticiones de *A*. Por tanto, al término del subprotocolo de finalización, el intercambio se asegura equitativo para las dos partes.

Como conclusión, el protocolo FPH cumple todas las propiedades enumeradas en II-A.

III. Descripción del Modelo de Análisis Automatizado

III-A. Coloured Petri Nets

CPN (Coloured Petri Net) [7] es un lenguaje de modelado que combina las características y propiedades de las Petri Nets ordinarias con un lenguaje de programación de alto nivel llamado ML (Modeling Language) [7]. Petri Nets aportan las primitivas para la interacción de procesos, mientras que por su parte, el lenguaje de programación proporciona la definición de los tipos de datos y la posibilidad de manipularlos. CPN puede ser usado como un método formal de análisis de sistemas distribuidos y protocolos de comunicaciones. Una red CPN es un modelo ejecutable que representa los estados de un sistema a lo largo de una línea temporal. Una CPN contiene cuatro tipos de componentes:

- Estados. Representan el estado del sistema en un instante determinado. Los estados cambian a partir de la activación de las transiciones.
- Transiciones. Se corresponden con una acción que hace cambiar el estado del sistema.
- Arcos. Son los enlaces entre los estados y las transiciones.
- *Color sets*. Son testimonios que van pasando a través de los estados y las transiciones, y que tienen un determinado valor, llamado color.

El estado global del sistema modelado, para cada instante después de la generación de un evento, determina lo que se le llama *marking*. Un *marking*, por tanto, es equivalente a una fotografía del estado del sistema tras cada evento que hace cambiar el mismo. Una de las herramientas que implementan CPN es CPNTools [7], la cual es usada en el desarrollo de la presente investigación. Se trata de una herramienta gráfica, que permite construir CPN de forma rápida y visual. Una vez el modelo está diseñado, se puede someter a simulación, proceso mediante el cual se genera el *state spaces*. *State spaces* es el conjunto de *markings* del sistema entre el evento inicial y el final. Por tanto, mediante este procedimiento se obtiene una definición completa del comportamiento del sistema a lo largo de su ejecución.

III-B. Suposiciones Iniciales y Metodología

Para usar CPN con el objetivo de modelar un protocolo, hay que definir una serie de consideraciones generales:

- Cada participante posee un identificador único.
- Cada participante conoce, como paso previo a la ejecución del protocolo, las claves públicas del resto de partes firmantes.
- Los algoritmos criptográficos usados son seguros.
- Los mensajes enviados entre la TTP y cualquier participante siempre son entregados al destino sin modificaciones (canal elástico)

La metodología a seguir para construir el modelo y analizar la equidad del protocolo es la siguiente:

- Construir el modelo:
  - Declarar los *color sets* para representar los mensajes y los elementos del protocolo.
  - Crear la red de alto nivel para modelar el conjunto del protocolo incluyendo los participantes.
  - Crear la red a nivel de entidad para modelar el comportamiento interno de cada participante.
  - Crear el nivel de proceso para cada nivel de entidad.
  - Declarar funciones y variables que serán usadas en el modelo.
- Generar el *state space*:
  - Definir el *marking* inicial para cada participante.
  - Generar el *state space* del modelo usando CPNTools.
- Crear las funciones para buscar estados de ataque.
- Extraer escenarios de ataque mediante la evaluación de los *markings* obtenidos.

III-C. Descripción del Modelo

El modelo, basado en el planteado por Sornkhom y Perpoontanalarp [6], define cuatro participantes: *A*(lice), *B*(ob), *I*(ntruso) y TTP. Mientras la TTP es estríctamente honesta, las otras partes pueden adquirir un rol malicioso. *A* y *B*, mediante sus roles maliciosos (A_m y B_m, respectivamente), pueden parar el intercambio o contactar con la TTP en diferentes etapas del protocolo, distintas a las definidas por el mismo, para intentar engañar a la otra parte. I es un participante malicioso que puede actuar como observador, capturando y almacenando los mensajes en tránsito, pero además puede realizar otras tareas: eliminar, reenviar o modificar mensajes en tránsito transmitidos por cualquier participante.

Para modelar los eventos de parada del intercambio realizados por cualquier participante malicioso (esto es, A_m, B_m o I), el modelo define un mecanismo para informar sobre estos eventos a la otra parte involucrada. Cuando ocurre un evento de este tipo, immediatamente se envía un mensaje desde el participante que realiza la eliminación del mensaje o la parada del intercambio a las otras partes implicadas. Con este procedimiento, se evita tener que usar nociones de tiempo (*timeout*) en el modelo. Cuando un mensaje de evento es recibido, el participante puede contactar con la TTP o bien parar el intercambio.

Otra consideración importante atañe al tipo de canal de comunicaciones usado para la intercambio de mensajes entre cualquier participante y la TTP, el cual por definición es elástico [12].

Con los datos proporcionados, creamos un escenario para ser usado en el modelado de un protocolo utilizando diferentes

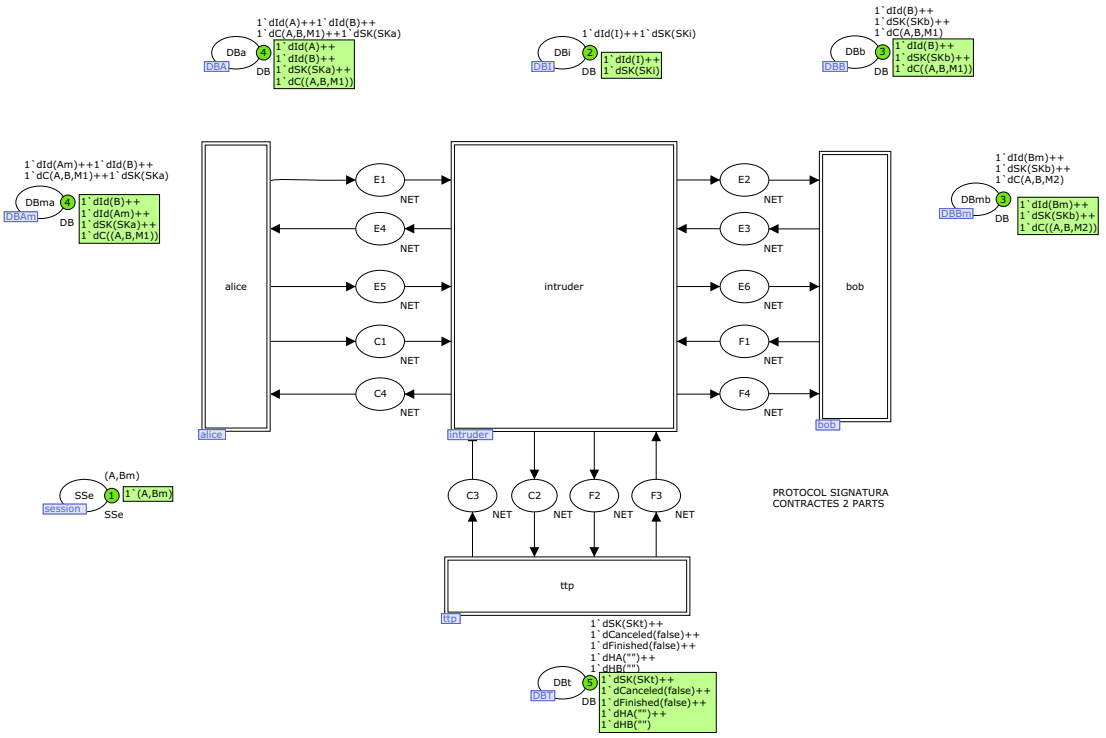


Figura 1. Esquema del nivel alto.

sesiones de ataque, donde cada una de ellas puede involucrar un iniciador (A o A_m) y un receptor (B y B_m). Notar que el I y la TTP estan presentes implícitamente en cada sesión. Por tanto, se pueden generar cuatro sesiones distintas combinando los roles de los participantes: (A, B), (A_m, B), (A, B_m) y (A_m, B_m) donde (X, Y) se corresponde con el participante que es el iniciador (X) y el receptor (Y), respectivamente. En este artículo no se consideran sesiones en paralelo, donde las partes maliciosas pueden participar en múltiples sesiones concurrentes, dejando esta tarea para próximos trabajos.

La arquitectura del modelo está claramente definida en tres grandes bloques o niveles: alto nivel, nivel de entidad y nivel de proceso. Todos los mensajes enviados por cualquier participante están compuestos por los identificadores de la fuente y el destino, así como el mensaje de protocolo incluído como *payload*.

El esquema del nivel alto (Figura 1) enseña la interacción entre todos los participantes involucrados en el protocolo, así como el flujo de mensajes entre los mismos. En el nivel alto se pueden consultar los contenidos de la base de datos de cada participante, las cuales contienen los mensajes de protocolo intercambiados. Finalmente, se puede observar y controlar el contenido de la sesión. La variable controla el rol de los participantes y cuáles son utilizados en la ejecución del protocolo (rol honesto o malicioso). Además, en la Figura 1 se puede observar como todos los mensajes son interceptados por I en su tránsito entre los participantes.

El nivel de entidad muestra con mayor detalle el flujo de trabajo de los participantes. En la Figura 2 se puede ver el nivel de entidad de A y sus dos roles. Las transiciones TA_1 a TA_4 se corresponden a su rol honesto, mientras que las transiciones TAm_1 a TAm_4 pertenecen al rol malicioso. La primera transición de A, TA_1 y TAm_1, genera el primer mensaje del protocolo y lo envía a B. Las transiciones TA_2 y TAm_2 reciben y verifican el segundo mensaje enviado por B y le retornan el tercer mensaje. TA_3 y TAm_3 tienen la responsabilidad de contactar con la TTP usando el subprotocolo de cancelación, y las últimas transiciones, TA_4 y TAm_4 reciben la respuesta desde la TTP. Notar que la selección de las transiciones que tienen que ser ejecutadas se lleva a cabo a través de la configuración de la sesión.

El nivel de proceso implementa todas las acciones que ejecutan los participantes y especifica la forma en la que las entidades se relacionan. Las acciones ejecutadas por cada proceso son atómicas, esto es, solo un proceso puede ser ejecutado al mismo tiempo y debe de terminar antes que el siguiente empieze su ejecución. Esto puede modelarse por un *token* único compartido por todos los partipantes. Éste es capturado por cada parte cuando un proceso empieza su ejecución y es liberado cuando el mismo proceso termina. Además, cada proceso está controlado por un mecanismo de control de flujo, mediante el cual un *token* es transferido a través de los participantes para cada paso del protocolo. Este *token* controla el orden en el que deben ser ejecutadas las acciones. Por ejemplo, el mecanismo controla que la generación de un mensaje se realice después de haberse ejecutado el proceso de verificación del mensaje préviamente recibido.

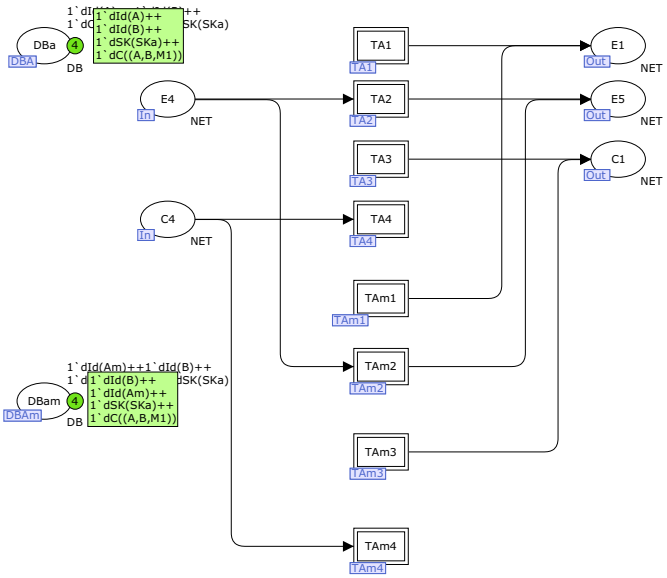


Figura 2. Nivel de entidad de *A* (a)

III-D. Funciones de Búsqueda

Para detectar y extraer escenarios de ataque a partir del *state space* generado por la utilidad, utilizamos una serie de funciones de búsqueda (Figura 3) para encontrar contenidos especiales en cada base de datos de los participantes. La función principal es *SearchCommitsTerminalNodes(ack,id)*, donde *ack* es el elemento o compromiso que se quiere buscar en la base de datos del participante identificado por *id*. La función está construida usando una función estándard de la utilidad CPNTools, llamada *PredNodes(p1,p2,p3)*. El primer parámetro de esta función es otra función capaz de examinar los contenidos de la base de datos de *id* buscando el elemento *ack*. El segundo parámetro es usado para elegir solo los nodos que sean terminales, es decir, nodos hoja del árbol generado, los cuales contienen una ejecución completa del protocolo. El último parámetro, *NoLimit*, indica a la función que debe de examinar todos los nodos y devolver todos los resultados posibles.

La función principal es usada para analizar la equidad del protocolo, aplicándola contra las bases de datos de los participantes del intercambio. La función devuelve una lista de nodos terminales, que contienen el compromiso buscado. El análisis de esta lista permitirá determinar si el intercambio es o no equitativo.

IV. Verificación Automatizada del Protocolo de Firma de Contratos FPH

IV-A. Evaluación de la Vulnerabilidad a Ataques ya Definidos

Con el paso del tiempo y las diversas investigaciones encaminadas a verificar los protocolos de firma de contratos, numerosos trabajos han encontrado ataques a estos protocolos. En esta línea, Bao [5] encontró tres ataques al protocolo ECS1 de Micali [2]. Posteriormente, Sornkhom y Permpoontanalarp encontraron dos nuevos ataques al mismo protocolo [6]. En todos los

```
fun SearchCommits( ack:DB, id:Id ) : Node list
  = PredAllNodes(
      fn n =>
      let
          val dba = Mark.Top'DBa 1 n
          val dbb = Mark.Top'DBb 1 n
          val dbi = Mark.Top'DBi 1 n
          val dbam = Mark.Top'DBma 1 n
          val dbbm = Mark.Top'DBmb 1 n
      in
          if (id=A) then
             cf( ack , dba ) > 0
          else if (id=B) then
             cf( ack , dbb ) > 0
          else if (id=Am) then
             cf( ack, dbam ) > 0
          else if (id=Bm) then
             cf( ack, dbbm ) > 0
          else (* id=I *)
             cf( ack , dbi ) > 0
      end
  )

fun SearchCommitsTerminalNodes( ack:DB, id:Id ) : Node list
    = PredNodes ( (SearchCommits(ack,id)) ,
                  fn n => (Terminal n) andalso (FullyProcessed n),
                  NoLimit)
```

Figura 3. Funciones de búsqueda de compromisos en las bases de datos de los participantes.

	$A \rightarrow B$:	$S_A(C, Z)$
	$B \rightarrow A$:	$S_B(C, Z), S_B(Z)$
IF (firmas de *B* son válidas)	$A \rightarrow B$:	M
IF (*B* recibe un *M* válido tal que $Z = E_{TTP}(A, B, M)$)		Intercambio completado
ELSE	$B \rightarrow$ TTP:	$A, B, Z, S_B(C, Z), S_B(Z)$
	TTP $\rightarrow A$:	$S_B(C, Z), S_B(Z)$
	TTP $\rightarrow B$:	M

Cuadro V
Definición del protocolo ECS1 de Micali

ataques, la vulnerabilidad afectaba a la equidad del intercambio, por los cuales, un participante podía obtener ventaja respecto del resto. Por esta razón, y como punto de partida para la verificación del modelo propuesto basado en CPN, se plantea evaluar el protocolo FPH para comprobar si es vulnerable a los ataques descritos al protocolo ECS1.

La definición del protocolo ECS1 de Micali (Cuadro V), ha sido adaptada para seguir la misma nomenclatura que el protocolo FPH. Además, se define $E_X(Y)$ como el cifrado del mensaje Y usando la clave pública de X. Para ECS1, A se comprometerá al contrato, C, como iniciador, si B posee los elementos $S_A(C, Z)$ y M aleatorio, tal que cumpla $Z = E_{TTP}(A, B, M)$. Por su lado, B se compromete a C como receptor si A posee $S_B(C, Z)$ y $S_B(Z)$.

Seguidamente, se describen cada uno de los cinco ataques al protocolo de Micali y se evaluará si son aplicables al protocolo FPH.

IV-B. Primer Ataque de Bao: A *es un iniciador malicioso y envía un elemento falso en el primer paso*

En el protocolo de Micali, el presente ataque (Cuadro VI) puede realizarse si A envia un elemento Z falso tal que $Z \neq E_{TTP}(A, B, M)$. En este caso, *A* siempre puede obtener el compromiso de *B* pero *B* nunca tendrá el compromiso de *A*. Este ataque es posible por el hecho que *B* no puede verificar los elementos recibidos en el primer paso.

La verificación automática del ataque se realiza configurando

A -> *B*:	$S_A(C, Z)$ donde $Z \neq E_{TTP}(A, B, M)$
B -> *A*:	$S_B(C, Z), S_B(Z)$
A -> *B*:	Nada
B -> TTP:	$A, B, Z, S_B(C, Z), S_B(Z)$
TTP -> *A*:	Nada
TTP -> *B*:	Nada

Cuadro VI
TRAZA DEL PRIMER ATAQUE DE BAO.

A → *B*:	$S_A(C, Z)$ donde $Z = E_{TTP}(A', B, M)$
B → *A*:	$S_B(C, Z), S_B(Z)$
A → *B*:	Nada
B → TTP:	$A, B, Z, S_B(C, Z), S_B(Z)$
TTP → *A*:	Nada
TTP → *B*:	Nada

Cuadro VII
TRAZA DEL SEGUNDO ATAQUE DE BAO.

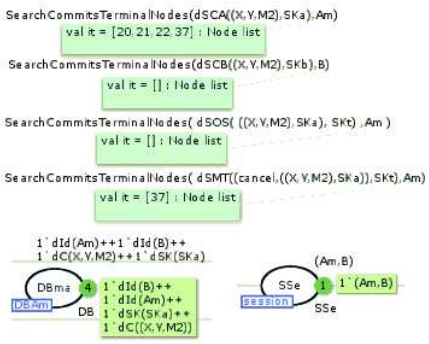


Figura 4. Resultados del primer ataque, contenido de la base de datos A_m y configuación de la sesión.

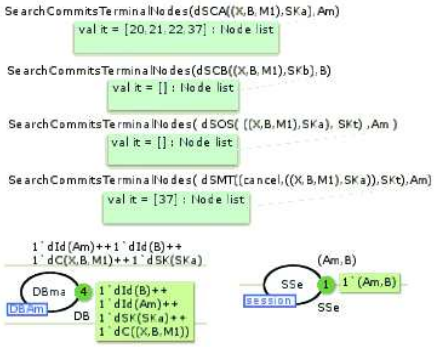


Figura 5. Resultados del segundo ataque, contenido de la base de datos A_m y configuación de la sesión.

el escenario con la sesión (A_m, B), haciendo que A_m envíe a B un contrato falso M_2 e imponiendo un iniciador (X) y un receptor (Y) arbitrarios. La primera función de la Figura 4 busca el primer compromiso, h_A en la base de datos de A_m, encontrándose en cuatro *markings*. La segunda función busca el mismo compromiso en B, y como puede observarse, no se encuentra. Esto es porqué B, después de verificar el mensaje recibido, determina que no es correcto y, por tanto, no lo guardará y no enviará respuesta. Por consiguiente, A_m, tal como se puede comprobar en la tercera función, no podrá poseer el segundo mensaje. La última función busca el elemento de cancelación enviado por la TTP a A_m. Por tanto, A_m tampoco puede obtener la evidencia NR desde la TTP.

Como se ha demostrado, FPH no es vulnerable al primer ataque, ya que *B* verifica los elementos recibidos en el paso 1. Si la verificación falla, *B* no envía el mensaje del paso 2 y entonces, *A* no puede enviar el mensaje 3. Si, por su parte, *A* intenta contactar con la TTP, ésta le enviará una prueba de cancelación y guardará *canceled=true*. *B* no puede contactar con la TTP ya que no dispone de ningún elemento válido recibido de parte de *A*.

IV-C. Segundo Ataque de Bao: A *conspira con otro iniciador* A' *y cambia su identidad en el primer paso*

En el protocolo de Micali, este ataque puede realizarse si *A* se confabula con un A' y envia un Z falso tal que $Z = E_{TTP}(A', B, M)$. En este caso, *A* maliciosa siempre puede obtener el compromiso de *B* sobre un contrato entre A' y *B*, pero *B* no puede obtener ningún elemento. El ataque es posible por el hecho que *B* no puede verificar la identidad de *A* en el elemento recibido en el primer paso.

El segundo ataque es verificado automáticamente en el modelo configurando la sesión con los mismos participantes que en el ataque previo, esto es (A_m, B), pero usando un contrato diferente. En este caso, el contrato contiene un iniciador falso (X), el receptor original (B) y el contrato préviamente acordado (M_1). Analizando los resultados de las funciones (Figura 5), se obtiene un resultado equivalente al primer ataque.

Por tanto, FPH no es vulnerable al segundo ataque, porqué *B* verifica los elementos recibidos en el paso 1, de modo que en caso de producirse el ataque, *B* no enviaría el mensaje del paso 2. Entonces, *A* no enviará el mensaje 3, el intercambio será parado y *A* no podrá obtener en ningún caso el compromiso de *B*. Si *A* intenta concluir el intercambio contactando con la TTP, recibirá de ésta una prueba de cancelación. Por su lado, *B* no contactará con la TTP ya que no quiere finalizar el intercambio porqué él conoce que el elemento enviado en el paso 1 es falso.

IV-D. Tercer Ataque de Bao: B *malicioso contacta con la TTP pidiendo la resolución de un contrato falso*

En el protocolo de Micali, este ataque puede llevarse a cabo si *B*, después de la recepción de un mensaje válido en el paso 1, contacta con la TTP para iniciar la resolución del intercambio. En esta petición, *B* incluye un contrato falso. En este caso, *B* malicioso siempre obtiene el compromiso de *A* sobre el contrato original, pero *A* obtiene el compromiso de *B* sobre el contrato falso, préviamente selecionado por *B*. El ataque es posible por el hecho que *A* no puede solicitar la resolución del intercambio original, obteniendo de la TTP, por contra, los elementos resultantes de la resolución iniciada por *B*.

El tercer ataque puede ser verificado usando el modelo aplicando una configuración de sesión compuesta por una *A*

$A \rightarrow B$:	$S_A(C, Z)$ donde $Z \neq E_{TTP}(A', B, M)$
$B \rightarrow$ TTP:	$Z, S_B(C', Z), S_B(Z)$ donde C' es un contrato falso
TTP $\rightarrow A$:	$S_B(C', Z), S_B(Z)$
TTP $\rightarrow B$:	M

Cuadro VIII
TRAZA DEL TERCER ATAQUE DE BAO.

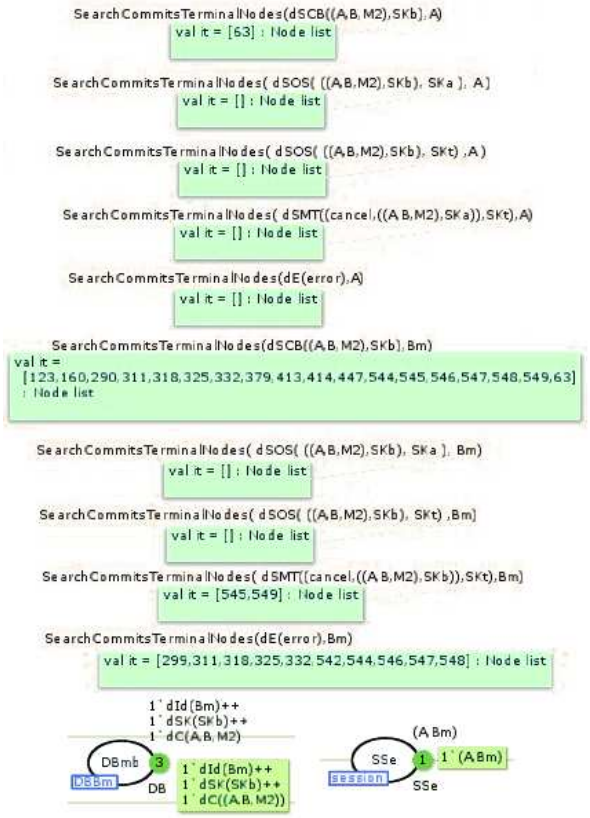


Figura 6. Resultados del tercer ataque, contenido de la base de datos B_m y configuación de la sesión.

honesta y un B_m malicioso, esto es (A, B_m). B_m construye un contrato falso (M_2) pero con los participantes originales (*A* y *B*). Así pues, en la Figura 6 se puede ver como *A* envía el primer mensaje correctamente a *B* y que este cambia su contenido, reenviando el segundo mensaje con el contrato modificado. Entonces, *A* detecta que el contrato ha sido manipulado y no envía el mensaje del paso 3. Por contra, *A* contacta con la TTP, pero usando el contrato original para cancelar el intercambio. Efectivamente, la TTP le remite un elemento de cancelación. Por su parte, B_m solo obtiene de la TTP elementos de cancelación y nunca el elemento NR.

Por consiguiente, el protocolo FPH se demuestra robusto frente el tercer ataque descrito, ya que cuando *A* recibe el mensaje falso del paso 2, esto es, $h'_B = S_B(M', A, B)$, *A* detecta el intento de ataque, parando el intercambio y contactando con la TTP. Si *B* contacta con la TTP en primer lugar y la petición contiene un h_B falso, la TTP es capaz de detectar

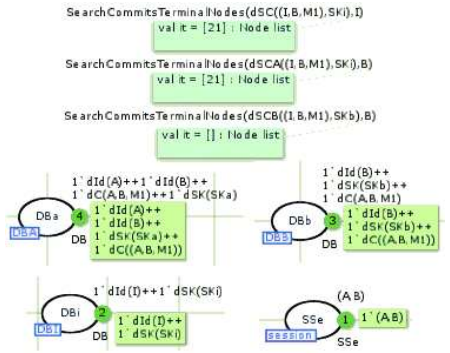


Figura 7. Resultados del cuarto ataque, contenido de la base de datos de los participantes y configuación de la sesión.

que los elementos h_A y h_B no se han obtenido del mismo contrato. Así, cuando *A* envía una petición de resolución, la TTP envía una prueba de cancelación, por tanto, el contrato no estará firmado. Si *A* contacta con la TTP en primer lugar, *A* obtendrá nuevamente una prueba de cancelación.

IV-E. Cuarto Ataque: Atacante captura el compromiso de B

El cuarto ataque descrito en [6] se basa en la incompleta definición del compromiso de *B* del protocolo ECS1. El mensaje $(S_B(C, Z), S_B(Z))$ representa la evidencia que prueba que *B* ha adquirido el compromiso sobre el contrato *C* con cualquier iniciador. La evidencia no está ligada al iniciador, por tanto, quien posea este elemento, puede afirmar que es el iniciador del contrato firmado por *B*.

Este ataque puede ser verificado en el modelo usando una configuración de sesión con dos participantes honestos, esto es (A, B). En este caso, hay que localizar estados en los que *I* haya capturado y modificado el primer mensaje (Figura 7). Por tanto, mediante la primera función, se encuentra un *marking*, en la que *I* intenta convertirse en el iniciador del intercambio, suplantando a A. En recepción, *B* verifica que ha recibido un elemento falso, y aunque lo guarda, no genera nunca la respuesta, h_B. Por consiguiente, *I* no obtiene el elemento 2.

Así pues, el protocolo FPH enlaza el compromiso de *B* con el contrato, hecho que impide la ejecución exitosa del cuarto ataque. La evidencia es el mensaje $h_B = S_B(M)$, aunque la obtención de este elemento no es suficiente para que alguien pueda probar que *B* se haya comprometido a si mismo con el contrato *M*. FPH especifica que *M* contiene el contrato a ser firmado e indica quién es el iniciador, *A*, y quién el receptor, *B*. Por esto, el protocolo FPH es resistente al ataque descrito en esta sección.

IV-F. Quinto Ataque: Intercambio de roles entre iniciador y receptor

En el ataque descrito en [6] al protocolo de Micali, una *A* maliciosa puede obtener el compromiso de *B* sobre un contrato entre *B* como iniciador y otra parte confabulada, A_r como receptor. Pero *B* no puede obtener nada. Para realizar este ataque, *A* construye un elemento *Z* falso con la identidad de

$A \rightarrow B$:	$S_A(C, Z)$ donde $Z = E_{TTP}(B, A_r, M)$
$B \rightarrow A$:	$S_B(C, Z), S_B(Z)$
$A \rightarrow B$:	Nada
$B \rightarrow$ TTP:	$A, B, Z, S_B(C, Z), S_B(Z)$
TTP $\rightarrow A$:	Nada
TTP $\rightarrow B$:	Nada

Cuadro IX
TRAZA DEL QUINTO ATAQUE DE BAO.

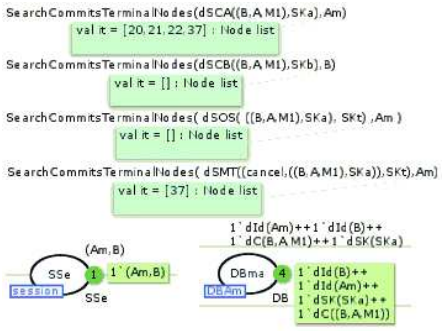


Figura 8. Resultados del quinto ataque, contenido de la base de datos B_m y configuación de la sesión.

B como iniciador y una A_r confabulada como receptor: $Z = E_{TTP}(B, A_r, M)$. Este ataque es posible por la inexistencia de un proceso de verificación de los elementos recibidos en el primer paso. Finalmente, *A* entrega $S_B(C, Z)$ y *M* a A_r. La TTP no puede enviar nada a *A* y *B* ya que el elemento *Z* no cumple con las especificaciones del protocolo. En este instante, A_r puede demostrar el compromiso de *B* sobre el contrato como iniciador, mientras que *B* no tiene ninguna clase de evidencia.

La verificación automática se realiza usando (A_m, B) como configuación de la sesión. En este caso, A_m cambia el contenido del contrato intercambiando los roles de los participantes sobre el contrato M_1. Los resultados de la aplicación de las funciones (Figura 8) muestran los mismos resultados obtenidos para el primer y segundo ataque. Por tanto, A_m no consigue en ningún caso obtener la evidencia de la firma del contrato ni de B ni de la TTP.

Como ha demostrado la verificación automatizada, FPH no es vulnerable a este ataque, ya que *B* verifica el elemento recibido en el paso 1 para comprobar su validez. Por esta razón, si *A* realiza cambios maliciosos en el mensaje, *B* lo detectará y ya no continuará con la ejecución del protocolo.

V. CONCLUSIONES

En el presente artículo hemos verificado formalmente el protocolo FPH [3], usando un nuevo modelo automatizado de análisis basado en CPN [7] y trabajos previos [6]. Hemos demostrado que el protocolo FPH no es vulnerable a los ataques descritos por Bao (3 ataques) [5] y los dos nuevos descubiertos por Sornkhom y Perpoontanalarp [6]. Paralelamente, hemos demostrado como puede ser utilizado el modelo automatizado para verificar protocolos de intercambio equitativo.

Con este trabajo, abrimos una línea en la que podremos usar el modelo para probar otras propiedades del protocolo FPH, como la verificabilidad de la TTP, así como abordar su aplicación a protocolos más complejos, como pueden ser los de tipo multiparte. Además, en próximos trabajos, adaptaremos el modelo para trabajar con nuevos escenarios de ataque, como ataques confabulados usando datos de múltiples sesiones concurrentes. En paralelo, se trabajará para incluir más control sobre el comportamiento del intruso y otras pequeñas mejoras.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el MEC y FEDER bajo los proyectos: ”Seguridad en la Contratación Electrónica basada en Servicios Web”(CICYT TSI2007-62986) y ARES ”Grupo de Investigación Avanzada en Seguridad y Privacidad de la Información”(Consolider - Ingenio CSD2007-004)

REFERENCIAS

[1] N. Asokan, M. Shunter, and M. Waidner, “Optimistic Protocols for Fair Exchange,” *4th ACM Conference on Computer and Communications Security*, pp. 7–17, 1997.

[2] S. Micali, “Simple and Fast Optimistic Protocols for Fair Electronic Exchange,” *Proceedings of 21st Symposium on Principles of Distributed Computing*, pp. 12–19, 2003.

[3] J. Ferrer-Gomila, M. Payeras-Capellà , and L. Huguet-Rotger, “Efficient Optimistic N-Party Contract Signing Protocol,” *Information Security Conference. 4th International Conference, ISC’01, LNCS 2200, Springer Verlag*, pp. 394–407, 2001.

[4] F. Thayer-Fábrega, J. C. Herzog, and J. D. Guttman, “Strand Spaces: Why is a Security Protocolo Correct?” 1998.

[5] F. Bao, G. Wang, J. Zhou, and Z. Zhu, “Analysis and Improvement of Micali’s Fair Contract Signing Protocol,” *Procedings of The 9th Australasian Conference on Information Security and Privacy*, pp. 176–187, 2004.

[6] P. Sornkhom and Y. Permpoontanalarp, “Security analysis of micali’s fair contract signing protocol by using coloured petri nets,” *9th ACIS Int. Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pp. 329–334, 2008.

[7] K. Jensen, L. M. Kristensen, and L. Wells, “Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems,” *Intenationals Journal on Software Tools for Technology Transfer*, pp. 213–254, 2007.

[8] J. Garay, M. Jackobson, and P. MacKenzie, “Abuse-Free Optimistic Contract Signing,” *CRYPTO’99, LNCS 1666, Springer Verlag*, 1999.

[9] N. Asokan, V. Shoup, and M. Waidner, “Asynchronous Protocols for Optimistic Fair Exchange,” *IEEE Symposium on Research in Security and Privacy*, pp. 86–99, 1998.

[10] J. Zhou, R. Deng, and F. Bao, “Some remarks on a fair exchange protocol,” *PKC 2000, LNCS 1751, Springer Verlag*, pp. 46–57, 2000.

[11] J. Ferrer-Gomila, M. Payeras-Capellà , and L. Huguet-Rotger, “Optimallity in asynchronous contract signing protocols,” *Trust and Privacy in Digital Bussines, TrustBus’04. LNCS 3184*, pp. 200–208, 2004.

[12] J. Zhou, R. Deng, and F. Bao, “Evolution of Fair Non-repudiation with TTP,” *ACISP99*, vol. LNCS 1587, pp. 258–269, 1999.

Stelin. Una herramienta pública para generación automática de estegotextos en lengua española.

Alfonso Muñoz Muñoz, Justo Carracedo Gallardo
Departamento de Ingeniería y Arquitecturas Telemáticas. DIATEL
Universidad Politécnica de Madrid. E.U.I.T Telecomunicación
Carretera de Valencia Km.7 – 28031. Madrid. España
{amunoz,carracedo}@diatel.upm.es

***Resumen*- La generación automática de estegotextos es una rama de investigación dentro de la esteganografía lingüística con un enorme potencial. Por desgracia, en la actualidad existen pocas investigaciones públicas sobre el potencial de este tipo de esteganografía en lengua española. En este artículo se centra el interés en la herramienta libre Stelin que implementa, en lenguaje JAVA, una variante mejorada de un algoritmo, de Peter Wayner, de generación automática de estegotextos pero aplicado a lengua española. La automatización y robustez de estos mecanismos facilitaría, se piensa, su utilización masiva en entornos web 2.0 y redes sociales, donde la información textual está por todas partes, facilitando comunicaciones anónimas.**

***Palabras Clave*- Stelin, esteganografía lingüística, estegotextos, generación automática, redes sociales, Peter Wayner.**

I. ESTRUCTURA DEL ARTÍCULO.

El presente artículo está estructurado en los siguientes apartados. El primer y segundo apartado es una breve introducción a los conceptos de esteganografía y estegoanálisis (donde se referencia los trabajos previos de los autores en esta temática), esta introducción permite comprender mejor el resumen de estado del arte de la esteganografía lingüística que se describe a continuación. El tercer apartado analiza las posibilidades de generación automática de estegotextos más conocidas. El cuarto apartado profundiza en la herramienta Stelin y en la generación de estegotextos basados en una variante del algoritmo de imitado estadístico de Peter Wayner. El último apartado sintetiza las conclusiones obtenidas y destaca posibles líneas de investigación futuras.

II. INTRODUCCION A LA ESTEGANOGRAFIA. TRABAJOS PREVIOS

La esteganografía es la ciencia y el arte de ocultar una información dentro de otra, que haría la función de *tapadera o cubierta*, con la intención de que no se perciba ni siquiera la existencia de dicha información [1]. En teoría, sólo quienes conozcan cierta información acerca de esa ocultación (un secreto) estarían en condiciones de descubrirla. En criptografía no se oculta la existen del mensaje sino que se hace ilegible para quien no esté al tanto de un determinado secreto (la clave). Por este motivo, los mensajes que se procuran ocultar usando técnicas esteganográficas, habitualmente, son previamente cifrados. La ocultación de mensajes usando procedimientos esteganográficos puede tener fines legítimos o ilegítimos, que pueden ser beneficiosos para proteger la privacidad de las comunicaciones o burlar censuras, o, por el contrario, ser vehículos para perpetrar actos criminales. Por estos motivos, en la presente década se está realizando una inversión importante en la detección de comunicaciones ocultas.

El estegoanálisis es la ciencia y el arte que permite detectar esa información oculta. En general, existen dos tipos de ataques estegoanalíticos: ataques activos y ataques pasivos. Los ataques activos se centran en la eliminación de la posible presencia de información enmascarada en un potencial estegomedio (cubierta o medio original que se utiliza para ocultar información). Estas técnicas son utilizadas especialmente para atacar algoritmos de watermarking. Por otro lado, los ataques pasivos se centran en el estudio de los potenciales estegomedios y la deducción de si almacenan información oculta. Los algoritmos estegoanalíticos más precisos (siglo XXI) son capaces no sólo de determinar el tamaño de la información oculta, sino de aplicar procedimientos de detección independientemente del conocimiento de la técnica de ocultación empleada; esto se denomina estegoanálisis a ciegas (blind steganalysis). El concepto de estegoanálisis a ciegas aprovecha el uso de diversos clasificadores (SVM, Fisher, etc) que mediante la definición de características propias de cada medio consiguen diferenciar entre cubiertas originales y potenciales estegomedios. No obstante si en el proceso de ocultación se toman las medidas de protección adecuadas, para el estegoanalista es inviable (según las publicaciones actuales) conseguir la extracción y recuperación de la información real enmascarada. En general, esta difícil tarea pertenecería a la ciencia del criptoanálisis.

A. Trabajo previo de los autores.

En 2005, en la línea de las investigaciones iniciadas por la comunidad científica, se inició un análisis de las posibilidades de detección de comunicaciones ocultas (estegoanálisis) en Internet. Algunos de los resultados obtenidos se hicieron públicos en 2007 con la publicación de la herramienta libre StegSecret [2]. Esta herramienta demuestra que es viable la automatización de la detección de información ocultada mediante muchos de los programas y técnicas

esteganográficas más difundidas. Estas investigaciones permitieron, paralelamente, observar la precisión de algunas de las técnicas de detección más precisas publicadas. De hecho, en general, los algoritmos de estegoanálisis tienen unos umbrales mínimos de detección que hacen que si se oculta una información de pequeño tamaño (mediante un mecanismo intrusivo) no se pueda distinguir su presencia con precisión (se producen falsos positivos que limitan la validez de la medida). En la práctica, cuanta menos información se oculte el impacto será menor, y la precisión de la medida de los algoritmos de estegoanálisis (la mayoría estadísticos) disminuirá. Esto hace que en la actualidad se utilicen mecanismos para reducir el impacto, por ejemplo el uso de matrices de codificación (el concepto de matriz de codificación hace referencia a procedimientos matemáticos que mejoran la relación información insertada-modificación de un estegomedio), la distribución de la información en varios portadores, así como técnicas para dificultar la recopilación de estegomedios a potenciales estegoanalistas (principalmente contenido diseminado en redes sociales). En 2008 generalizamos estos aspectos en su relación con las nuevas formas de comunicación colaborativa (redes sociales, redes p2p, p2m, etc) y se analizó el potencial esteganográfico de la distribución de información oculta multiproveedor y multiportador [3][4].

Todo este conocimiento se focaliza en la línea actual de investigación centrada en el potencial de las redes sociales para enmascarar información oculta. En este tipo de redes la posibilidad de utilizar estegomedios basados en lenguaje natural es de gran interés por su difusión. Este artículo se circunscribe en los análisis iniciados a este respecto.

III. ESTEGANOGRAFIA LINGÜÍSTICA.

En las últimas décadas el avance en procedimientos de ocultación y técnicas de estegoanálisis se centra especialmente en el contenido multimedia como estegomedio, entre otras cosas por su presencia masiva en la sociedad del conocimiento actual. No obstante, existe aún un medio todavía más masivo. La información textual (lenguaje natural) está presente en todo, Internet y las redes sociales es una buena muestra de ello, y este hecho lo hace lo suficientemente atractivo para considerarlo como un potencial estegomedio. La cantidad de información textual y su distribución global puede ser una ventaja para dificultar la capacidad de un estegoanalista en "separar el grano de la paja".

La consideración de la información textual como un potencial estegomedio no es ni mucho menos nueva. A lo largo de los siglos se han documentado múltiples formas de ocultación en soportes y formas varias [5]: cartas, libros, telegramas, poesías, canciones, revistas, periódicos (por ejemplo, *newspaper code* en la época Victoriana o la verja de Cardano en el siglo XVI); o más recientemente en canales de mensajería instantánea (messenger, IRC), utilizando el "ruido" de las traducciones automáticas, basándose en lenguajes de marcado (HTML y XML), etc [2]. Es común ver una clasificación clásica de estas técnicas en términos de *open codes y semagrams*, en terminología inglesa. Los *open codes* genéricamente se refieren a textos de apariencia inocente, que ocultan información recuperable utilizando ciertas letras, palabras, frases del texto o comunicación (métodos basados en esto son: Cues, Null Ciphers, Jargon Code y Grilles), mientras que los *semagrams* son el conjunto de técnicas que consisten en la utilización (variación) de la estructura y formato de los elementos de un texto, aunque visibles, no por ello son fáciles de detectar [5].

En general, las técnicas de ocultación basadas en información textual se basan en la utilización de textos existentes (se modifican) o la creación de textos de forma automática. La seguridad de estos estegotextos debe ser analizada desde diferentes puntos de vista, considerando ataques lingüísticos (sintáctico, semántico y de coherencia) por parte de máquinas y analistas, y ataques puramente estegoanalíticos y estadísticos (análisis de entropía, análisis de frecuencia de caracteres-palabras, ataques basados en conocimiento de cubierta original y cubierta modificada, etc).

En la práctica, el lenguaje natural, como estegomedio, es muy poco redundante (ruidoso) en comparación con otros estegomedios como son las imágenes o los videos, lo cual hace más complicado la creación de algoritmos robustos de ocultación de información en lenguaje natural (información textual), lo que hace que las técnicas de ocultación requieran una gran cantidad de información textual para ocultar una cantidad de información no muy abultada. Por este motivo, todo mecanismo que ayude a reducir el impacto en un texto al ocultar una información es importante. A recursos ya comentados (matrices de codificación y distribución) se le pueden sumar todos aquellos procedimientos que compensen las perturbaciones introducidas en la cubierta origen, o ideas más interesantes como es la negación plausible [6].

Por todo esto, en la presente década se han documentado esfuerzos notorios en la robustez de algoritmos esteganográficos utilizando el lenguaje natural en idiomas tan dispares como: inglés, ruso, mandarín, coreano, persa, etc.

En lengua española, en general, no existen estudios avanzados de esteganografía lingüística, hasta lo que tenemos constancia, salvo alguna notoria excepción [7]. Esto abre un enorme potencial de investigación y experimentación.

En la práctica si se desea profundizar en esta temática para lengua española debe considerarse, en la actualidad, las siguientes líneas de investigación para ocultar información en lenguaje natural [8]:

a) **Generación automática de estegotextos** (se analiza en detalle en los siguientes apartados).

b) **Generación de estegotextos basados en la modificación de textos existentes.** Dentro de esta última alternativa existen diferentes recursos de interés: **modificaciones léxicas** (por ejemplo, basadas en sustitución de sinónimos), **modificaciones sintácticas y semánticas** (por ejemplo, transformación activa/pasiva, desplazamiento en las posiciones de los verbos, ontologías aplicadas a esteganografía lingüística, etc), **ocultación basada en la posibilidad de traducir una sentencia** en varias equivalentes en otro idioma, **ocultación basada en la modificación del formato-estructura** de un texto (espacios de tamaño variable entre palabras, cambios sucesivos de: la fuente, color, tamaño de letra, mayúsculas, etc), **ocultación basada en errores tipográficos, ortográficos, abreviaturas y símbolos de puntuación**, etc.

En los siguientes apartados se profundiza en la primera línea de investigación. Se analiza, dado que existe poca documentación al respecto, las ventajas e inconvenientes de alguno de los procedimientos de generación automática de estegotextos más conocidos en su aplicación a lengua española.

III. GENERACIÓN AUTOMÁTICA DE ESTEGOTEXTOS.

Los mecanismos tradicionales para ocultar información en lenguaje natural consisten en la modificación de textos o documentos existentes. El problema fundamental de esta modificación reside en el peligro de que un potencial estegoanalista pudiera conseguir el texto original sin modificar y realizar ataques clásicos de comparación estegotexto – texto original, lo que delataría fácilmente la presencia de información oculta. Ante este hecho, una solución posible es la realización de modificaciones sobre un estegotexto creado automáticamente a la medida, e incluso único por cada comunicación.

En las últimas décadas, las dos propuestas más interesantes de generación de estegotextos de forma automática se basan en los trabajos de Peter Wayner en la década de los 90 [9].

Un trabajo interesante consiste en un algoritmo de generación automática de estegotextos basado en la imitación estadística de una o más fuentes de textos. Este algoritmo y su automatización se analizará más profundamente en el siguiente apartado con la herramienta pública Stelin.

El segundo trabajo destacable de Wayner consistió en hacer útil las construcciones CFGs (Context-Free-Grammar) en esteganografía lingüística, derivando principios de la teoría de la gramática generativa propuesta por el lingüística A.Noam Chomsky en la década de los 60.

En la práctica, una CFG puede ser vista **como un sistema combinatorio discreto que permite construir infinitas frases a partir de un número finito de elementos** mediante reglas diversas que pueden formalizarse mediante una gramática formal gobernada por normas de transformación (una CFG se compone de terminales, variables y producciones). Wayner trabajó en la posibilidad de utilizarlas en esteganografía para la generación automática de estegotextos ya que al menos tendrían validez gramatical-sintáctica (al menos para lengua inglesa). La ocultación de información se realizaría mediante la selección de elementos concretos dentro de una regla específica, regla que sería elegida mediante algún algoritmo de selección concreto.

Variable_Inicio S ::= AB (.5) | AC (.5)
A ::= “Hola”(.5)|“Que tal”(.5)
B ::= “sabes si” C (.5) | “te han dicho si” C (.5)
C ::= “Juan” D (.125) | “Pedro” D (.125) | “Lucas” D (.125)|“Tomas”D(.125)|“Irene”D(.125)|“Sandra”D(.125)|“ Marta” D(.125)| “Bea”D(.125).
D ::= “va a venir a jugar al” E (.5) | “va a llegar al partido de” E (.5)
E ::= “fútbol” F (.25), “baloncesto” F(.25) , “tenis” F(.25), “ping-pong” F(.25).
F ::= Un beso (1).

Fig. 1. Ejemplo de PCFG en lengua española (en formato BNF).

Un estegotexto ejemplo de la regla de la Fig.1 (oculta 8 bits) sería: “*Hola sabes si Irene va a venir a jugar al tenis. Un beso”*. Wayner desarrollo varios ejemplos interesantes aplicando estas ideas: spammimic, baseball game, etc [9].

Aunque Wayner, fue uno de los primeros que se esforzó en formalizar la construcción de CFGs seguras con utilidad esteganográfica y analizar su seguridad [9], es cierto que su utilidad esteganográfica debe ser muy matizada. Entre los problemas a analizar es destacable el problema de la privacidad de la gramática.

La calidad (sintáctica, semántica y de coherencia global) de un estegotexto generado depende de la gramática utilizada y esta, en principio, debe permanecer secreta y compartida entre emisor y receptor. Una gramática construida manualmente proporcionaría una calidad de estegotexto muy elevada a costa de un proceso tedioso de generación y la creación de un enorme volumen de reglas para evitar la probable repetición de frases y términos que facilitarían la tarea de un estegoanalista, con el riesgo de si la gramática es comprometida (a efectos prácticos actúa como clave) se debería repetir el proceso tedioso de generación. Aunque el procedimiento de generación de las reglas se simplifique mediante algún proceso automático (por ejemplo, de imitado de la estructura gramatical de uno o más textos) deben considerarse otras cuestiones. Por un lado, las palabras (términos) en una CFG se relacionan con sus vecinos en formas fijas. Aunque se añadan modelos estadísticos para seleccionar las reglas y términos involucrados en una gramática (Probabilistic Context-Free-Grammars –PCFG-), para dificultar ataques de análisis, siempre existirán correlaciones mutuas si se quiere que el texto sea legible por un humano [9]. Por otro lado, deben considerarse ataques basados en estudio de terminales (información última de cada regla), ya que aunque las variaciones de texto creados puedan crecer sustancialmente con el tamaño de una gramática dada, el número de terminales está limitado por el tamaño de la gramática, lo cual significa que forzosamente, si el texto es lo suficientemente grande, combinaciones lineales de terminales se tienen que producir y por tanto repetir.

En resumidas cuentas, resulta realmente complejo utilizar CFGs en herramientas públicas de manera robusta en la concepción actual, sobre todo si se considera que la gramática no es privada o al menos no compartida entre emisor y receptor. Alguna propuesta, en este sentido, ha sido documentada, por ejemplo, la herramienta NICETEXT [10] que crea estegotextos basados en PCFGs que se crean dinámicamente de fuentes de entrenamiento (textos) y esta gramática no necesita ser compartida entre emisor y receptor, ya que el proceso de recuperación no requiere del conocimiento de la misma. A pesar de todo, en 2008 la comunidad científica china publicó, una vez más [11][12][13][14], una serie de avances en estegoanálisis lingüístico que ponen de manifiesto múltiples consideraciones adicionales a tener en cuenta. Sus ataques [15] a NICETEXT y otras herramientas (TEXTO y basadas en cadenas de Markov) se basaron en la suposición que en un texto natural las palabras se distribuyen de manera no equitativa, es decir, algunas palabras se repiten frecuentemente en algunos lugares pero rara vez en otros, es decir, aplicar estadística y principios de localidad en su uso al estegoanálisis lingüístico (por ejemplo, mediante el cálculo

de estimadores que entrenarán un clasificador Support Vector Machine que diferenciará entre textos sin información oculta y estegotextos). Los resultados publicados [15], a falta de ser contrastados con otros estudios independientes, indican que el ratio de detección excede del 90% para estegotextos en lengua inglesa de tamaño en torno a 5KB.

En resumen, todas las características analizadas deben ser tenidas en cuenta para desarrollar herramientas públicas robustas basadas en la imitación de gramáticas, especialmente, en nuestro caso, si se quiere analizar la seguridad esteganográfica de estos principios en lengua española.

No obstante, dado los múltiples problemas de base anteriores a solucionar, la investigación se inicia analizando algo más en profundidad en su aplicación a lengua española otra de las propuestas formuladas por Peter Wayner en la década de los 90, y posibles mejoras de esta.

IV. STELIN. UNA HERRAMIENTA PÚBLICA DE GENERACIÓN AUTOMÁTICA DE ESTEGOTEXTOS EN LENGUA ESPAÑOLA.

En las últimas décadas los esfuerzos en generación automática de estegotextos se centran, principalmente, en la generación de estegotextos que imiten la gramática (sintaxis) y la estadística de un texto "típico" en una lengua concreta.

Peter Wayner en la década de los 90 publicó un procedimiento de generación automática de estegotextos (T) basado en el imitado estadístico de una o más fuentes de textos (S) que resulta interesante analizar [9]. La idea es sencilla: "Cójase una función de imitado f que modifique un fichero A de forma que asuma las propiedades estadísticas de otro fichero B. Es decir, si p(t,A) es la probabilidad de que una cadena t suceda en A, entonces una función de imitado f, hace que la p(t,f(A)) sea aproximadamente p(t,B) para toda cadena t de tamaño menor que n". La complejidad del modelo estadístico de imitado (análisis de frecuencia) depende, precisamente del orden estadístico n (orden de complejidad del algoritmo). Según está idea, Wayner definió el siguiente algoritmo de imitado:

1. Constrúyase una lista de todas las diferentes combinaciones de n letras que ocurran en S y contabilícese el número de veces que ocurren en S.

2. Elegir una de ellas aleatoriamente que actuará de semilla inicial. Esto generará las primeras n letras de T (el estegotexto).

3. Repetir este punto hasta que se genere todo el texto deseado:

a. Cójase las n-1 letras siguientes de T. Buscar en la tabla estadística (creada) todas las combinaciones de letras que comienzan con esas n-1 letras.

b. La última letra de esas combinaciones forma el conjunto de posibles elecciones para la siguiente letra que será añadida a T.

c. Elegir entre esas letras y usar la frecuencia de sus ocurrencias en S para "evaluar" cuál es la mejor elección.

d. Añadirla a T.

Por ejemplo, un primer orden de imitado genera caracteres aleatorios de acuerdo a la distribución estadística del texto de entrenamiento. En un segundo orden imita la distribución de parejas de caracteres de los textos S de entrenamiento, y así sucesivamente para órdenes mayores. El proceso de ocultación de información se realiza mediante la selección de las opciones de la próxima letra a mostrar. Wayner justificó como esto se podría hacer, entre otras opciones, utilizando un árbol de Huffman, que basándose en las frecuencias de aparición de los caracteres (por ejemplo) les asignaría un código (código que se utilizará para ocultar una información). Si la selección de las ramas de este árbol (que imita la estadística a la fuente), es aleatoria el texto resultante imitará (o se aproximará) a la distribución estadística del texto fuente (ver Fig.2).

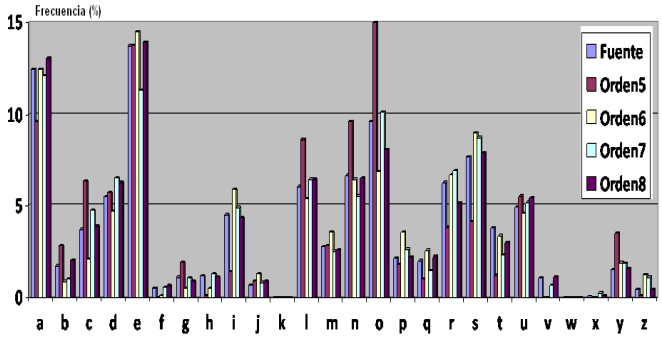


Fig. 2. Ejemplo de comparación de la distribución de frecuencias de caracteres de una fuente de texto de entrenamiento (los 13 primeros capítulos del Quijote con un total de 167.313 caracteres) y un estegotexto generado de ocultar 256 octetos (2048 bits) de información oculta a partir de dicha fuente (como nivel de atomicidad el carácter).

Se supone (por la información publicada [9]) que para lengua inglesa, dependiendo del texto y del orden (texto de al menos decenas de KB y orden mayor que 8), pueden obtenerse estegotextos con validez léxica y sintáctica, e incluso con apariencia semántica-estructural. En lengua española, por las pruebas iniciales realizadas (Stelin implementa, además, el algoritmo original de Peter Wayner), esta afirmación no puede mantenerse en general, y el resultado depende mucho de la fuente de entrenamiento elegida. En la práctica, la ocultación de unas pocas decenas de octetos producirá estegotextos con algún error léxico, gramatical o de repetición de términos (ejemplos Fig.3 y Fig4).

"que no cese la producción y sus políticos. La seguridad y soberanía y la seguridad y soberanía alimentaria desde una posición de defensa campesinas de los poderes económicos, sociales, económicas y culturales. E) El rechazo social a las políticas de producir riqueza, el capitalista de la soberanía y la seguridad alimentaria en el mercado mundial, así como su reconocimiento recíproco como sujetos de derechos de la pobreza y la exclusión. Hoy no se producen los alimentos), homogeneizando culturas, criterios y técnicas productiva y culturales, manteniendo el control de los poderosos, el incremento de la des"

Fig. 3. Ocultación de 10 octetos de información (0xF0, 0x0F, 0xAA, 0x71, 0xF0, 0x0F, 0xAA, 0x71, 0xF0, 0x53) utilizando un texto de 24KB (Seguridad Alimentaria y sus condiciones de posibilidad. http://www.kaos

enlared.net/noticia/seguridad-alimentaria-condiciones-posibilidad) y orden de complejidad 11. Expansión 1:61. Modo carácter.

[INICIO TEXTO] *qué dulcísimo sueño! Calderón. Al brillar un relámpago y rujo en la tormenta. Yo río en los alcores susurro en la alta hierba, suspiro yo. Hoy la tierra. Yo soy nieve en las cumbre, soy fuego en las arenas, azul onda en los mares y espuma en las tumbas y en las ruinas hiedra. Yo atrueno en el torrente, y silbo en la centella y ciego en el relámpago y rujo en la tormenta. Yo río en los alcores susurro en la alta noche a los maitines llama! ¡Cuántas veces el genio así duerme en la rama esperando la mano de nieve que sabe arrancarme del mísero suelo, y anegarme en su luz, y con ella en lumbre encendido fundirme en un beso dos almas confundidas; mientras responda el labio risas que se desmienten con los ojos los párpados, dormida, tranquilo fulgor vierten, cual derrama de luz, templado rayo, lámpara transparente. ¡Duerme! Despierta, tiemblo al mirarte; dormida, me atrevo a verte; por eso, alma de mi alma, y una voz, como Lázaro, espera que levantaba el palpitante seno, una flor se mecía en compasado y tenue, escucho yo un poema de ternura infinita. Ella tiene la luz, tiene el perfume, el color y la ternura, del bardo inglés en el horrible drama, la dulce Ofelia, la razón perdida cogiendo flores y cantando la eterna canción y, al juntarse allí en el cielo, y les he visto el fin o con los --átomos del aire tiemblan. Es tu boca de rubíes purpúrea granada abierta que en la primavera; entre sus pestañas semejan* **[FIN TEXTO]**

Fig. 4. Ocultación de 10 octetos de información (0xF0, 0x0F, 0xAA, 0x71, 0xF0, 0x0F, 0xAA, 0x71, 0xF0, 0x53) utilizando una versión digital de la obra RIMAS (42 KB) de Gustavo Adolfo Bécquer. Orden de complejidad 11. Expansión 1:142. Modo Carácter

La idea de Peter Wayner podría ser mejorable (o eso se piensa) si se considera un nivel de atomicidad de entrenamiento diferente, por ejemplo, la utilización de la palabra en lugar del carácter, dado que se espera que los resultados mejoren, al menos sintácticamente (y se eviten errores léxicos). Otro nivel atómico sería posible pero esto condicionaría el factor de expansión y la capacidad de ocultación, por ejemplo un párrafo, un verso, etc.

La herramienta libre Stelin, implementada en lenguaje JAVA, permite generar automáticamente estegotextos en lengua española basada en estos criterios y algunas mejoras adicionales (http://steling.sourceforge.net).

El algoritmo implementado en Stelin para imitar una fuente de texto de entrenamiento, considerando como nivel de atomicidad la palabra es el siguiente:

1. El proceso de generación se basa en el análisis de bloques de n palabras, extraídas del texto de entrenamiento, mediante una ventana deslizante que se desplaza una posición para cada nuevo bloque. Es decir, el primer bloque tendrá los términos de 0 a n-1, el segundo bloque de 1 a n, y así sucesivamente.
2. N define el orden de complejidad del algoritmo, lo que significa el número de palabras a considerar consecutivamente. Por tanto, su aparición viene condicionada por la aparición de palabras que le preceden o suceden.
3. Las palabras se relacionan mediante nodos enlazados en los que se contabiliza el número de veces que se han repetido en el texto de entrenamiento. Según esto, existirá una tabla raíz que almacenará todas las "palabras diferentes" que existan en el texto fuente.

Basado en lo anterior, el algoritmo de generación de estegotextos funcionaría, en general, de la siguiente manera:

a) Se selecciona una "palabra" aleatoriamente de la tabla raíz (podría considerarse otro criterio con fines sintácticos, por ejemplo, hacer que el texto empezase por un artículo). De esta forma, para un mismo texto de entrenamiento se podrían obtener diferentes estegotextos.

b) Si esta palabra no tiene sucesores (no apunta a otro nodo), se elije otro término de la tabla raíz (paso a). Si el nodo sucesor solo tiene una palabra, esta palabra se añade al estegotexto (no es posible ocultar información en este caso) y se elige el siguiente nodo disponible. Si el nodo sucesor tiene varias palabras posibles entre las que elegir se elije aquella cuya rama del árbol de Huffman, generado de las posibles palabras (y sus frecuencias), coincida con la información a ocultar, y se elige el siguiente nodo disponible.

c) Si se llega al último nodo (orden n=8, por ejemplo, 8 palabras consecutivas) se elige la última palabra seleccionada para el estegotexto y se vuelve al paso b). Este proceso se repite hasta que se genere el estegotexto que oculta la información deseada.

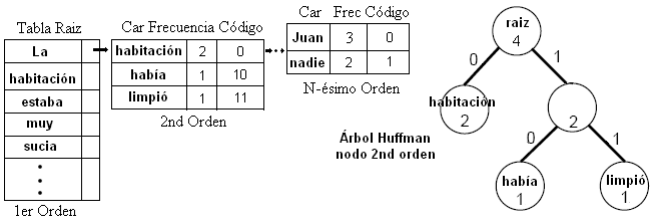


Fig. 5. Variante del algoritmo de P. Wayner . Atomicidad = Palabra.

Esta variante (que se ilustra en la Fig.5), genera estegotextos de mayor tamaño (a mayor nivel de atomicidad es más probable que los elementos no tengan tantos sucesores diferentes), pero es más fácil obtener textos con validez léxica y sintáctica, e incluso, en ocasiones, con apariencia semántica.

[INICIO TEXTO] *pesadilla. Está el sol en el ocaso. Suena el agua clara no mitiga, la amargura del tiempo de mentira, de infamia. A España toda, la luna llena, el ojo encandilado del búho insomne sueño mío! ¡Este frío de un amanecer en la tierra, y en este nuevo ejido sin duda, el amor a mujer el que llevó a un límite infranqueable la desubjetivación del sujeto. "¿Y cómo no intentar —dice Martín— devolver a mi oído, por la ventana de mi estancia, iluminada por esta luz invernal, —la tarde gris de plomo y azul de plata, con manchas de roja herrumbre, todo envuelto en luz violada. ¡Oh tierras de Alvargonzález, en el corazón de una tarde inmensa; mas falta el hilo entre los dos! Al borrarse la nieve, se alejaron los montes de la sierra. La tarde está cayendo frente a los caserones de sus lares; la tempestad llevarse los limos de una manera española, que fue casarse con una tarde clara y amplia como el hastío, cuando el eje del planeta se vence hacia el poeta admira y calla, el sabio mira y la noche azul ardía toda sembrada de estrellas. ¡Padre!, gritaron; al fondo de la laguna serena cayeron, y el eco ¡padre! repitió de peña denegrida, vuelve mi corazón a*

su faena, con el viento... ¡el viento de la tarde en su **[FIN TEXTO]**

Fig. 6. Ocultación de 16 octetos (0xF0, 0x0F, 0xAA, 0x71, 0xF0, 0x0F, 0xAA, 0x71, 0xF0, 0x0F, 0xAA, 0x71, 0xF0, 0x0F, 0xAA, 0x71). Texto Fuente obra "Poesías Completas" de Antonio Machado (290KB texto plano, versión digital). Orden de complejidad 9. Expansión 1:76. Modo Palabra.

[INICIO TEXTO] *viento... El viento traía perfume de rosas, dolor de campanas... Doblar de campanas lejanas, llorosas, suave de rosas aromado aliento ... ¿Dónde están los huertos floridos de la fuente sueñan... Sí, te conozco, tarde alegre y en el hogar campesino armó la envidia pelea. Casáronse los mayores; tuvo Alvargonzález nueras, que le trajeron cizaña, llenan la tierra maldita, tenaz a pico y a la tarde de abril que moría: ¿Al fin la sombra del sendero y el agua del mesón en el azul lejano. De tu morena gracia, de tu sombra a un hombre pensativo y a un agua de la peña? El hombre es por natura la bestia paradójica, un animal absurdo que necesita lógica. Creó de nada un mundo y, su obra terminada, "Ya estoy en el secreto —se dijo—, todo es la laguna insondable. Un buhonero, que cruzaba aquellas tierras errante, fue en Dauria acusado, preso y muerto en el aire ha abierto, y una mata de espliego castellano lleva en el pico a tu jardín diserto —mirto y laureles— desde el alto llano en donde el ojo alcanza su pleno mediodía (un diminuto bando de cuervos enronquece en busca de su peña denegrida, vuelve mi corazón a su faena, con néctares del campo el agua clara corriendo, mientras los dos asesinos tienen la maldición en sus campos. Ya el pueblo impío que juega al mus, de espaldas a la mar y la llanura, caminata o singladura, siempre larga, diéronle, para su prosa, viento recio, sal amarga, y el eco duerme, rodea; agua clara donde beben las rosas blancas, y ante el blanco lino que en los labios ... De tu mirar de sombra quiero llenar mi vaso. Para tu linda hermana arrancaré los montes sin nieve son de violeta. La tierra de mi corazón. Di, ¿por qué acequia escondida, agua, vienes hasta mi, manantial de nueva vida de donde nunca vivida* **[FIN TEXTO]**

Fig. 7. Ocultación de 16 octetos (0xAA, 0x71, 0xF0, 0x0F, 0xAA, 0x71, 0x54, 0x72, 0xAA, 0x71, 0x28, 0xF5 ,0xAA, 0x77, 0x00, 0xAC). Texto Fuente obra "Poesías Completas" de Antonio Machado (290KB texto plano, versión digital). Orden de complejidad 10. Expansión 1:109. Modo Palabra.

La Fig.6 y Fig.7 recogen unos ejemplos de estegotextos generados automáticamente en lengua española. En estos puede observarse, aplicando el algoritmo de la herramienta de Stelin descrito, que los estegotextos no finalizan necesariamente con una estructura puramente sintáctica. Esto puede solucionarse, por ejemplo, mostrando términos (ocultando una información aleatoria de relleno) hasta encontrar un fin de cadena (por ejemplo, un punto). Este hecho facilitaría a su vez la ocultación de una pequeña cantidad de información, si se desea, basada en principios de negación plausible [6].

En cualquier caso, la selección de los textos de entrenamiento y orden de complejidad son vitales para la generación de estegotextos de calidad (véase Fig.3,4,6,7). Diferentes tipos de textos podrían ser considerados como fuente para ocultar información (poemas, novelas, artículos periodísticos, código de programación, etc). Si el texto fuente es más grande es más probable que existan diferentes alternativas que sucedan a una palabra y por tanto la capacidad de ocultación sea mayor (piénsese en lengua española por ejemplo en la presencia de preposiciones y determinantes). Desde un punto de vista lingüístico deberían, al menos, evitarse o filtrase fragmentos de texto que claramente afecten a la coherencia en los estegotextos creados (ver Fig.3). Entre estos, índices, títulos, numeraciones (a), b), c), I, II, III), fechas, referencias, etc.

Por otro lado, a falta de una mejor formalización, las pruebas realizadas indican, que en general, un orden 8 o superior proporciona unos resultados léxicos y sintácticos razonables. La cuestión está en determinar si el esfuerzo merece la pena, si para un texto dado un orden N+1 es mejor (estadística y lingüísticamente) que un orden N o menor, ya que a mayor orden es más probable que el factor de expansión sea mayor y el estegotexto generado (más grande) sea más "atacable".

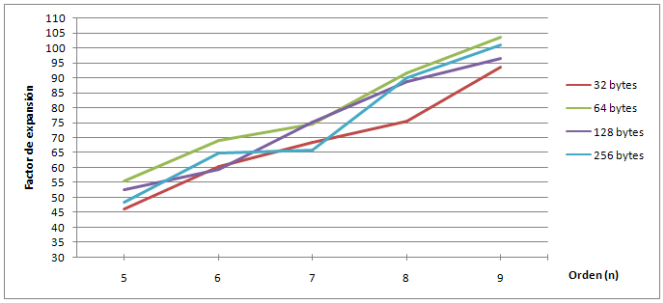


Fig. 8. Relación orden de complejidad y factor de expansión (mensaje a ocultar-estegotexto creado) para diferentes mensajes a ocultar de pequeño tamaño. Texto original: versión digital de Poesías Completas (290KB) de Antonio Machado (51.531 palabras).

Las pruebas actuales indican que el algoritmo implementado sólo permitiría la ocultación de una breve cantidad de información (decenas de octetos) con ciertas garantías léxicas, gramaticales e incluso de apariencia semántica (coherencia). No obstante, las pruebas siguen abiertas para intentar identificar si ciertos textos fuentes son más indicados para ocultar mayor cantidad de información.

Con la capacidad de ocultación actual, unas pocas decenas de octetos, se podría intercambiar breves mensajes de información, urls o claves criptográficas. Por ejemplo, una información de 16 octetos (128 bits) podría codificar, mediante un alfabeto de 32 elementos (27 letras y 5 símbolos adicionales), hasta 25 letras. Por ejemplo, un mensaje de movilización como: "a las doce en la plaza".

A. Problemas estadísticos del algoritmo de Peter Wayner y variantes.

Los estegotextos generados deben ser analizados no sólo lingüísticamente, sino además, con toda una serie de análisis estadísticos y estegoanalíticos. El ejemplo más clásico son los estudios basados en la estadística. En la práctica la aproximación estadística de la fuente de entrenamiento (texto) realizada por la idea de Wayner y variantes dependerá de varios factores, entre otros de la función de imitado utilizada.

La implementación actual de Stelin utiliza como función de imitado el algoritmo de Huffman (al igual que la idea original de Peter Wayner), por ser una buena elección, de esta manera, una posible codificación para 3 elementos (a,b,c) con

probabilidades (0'80, 0'13,0'07) sería (1,01,00). Si su función inversa es usada como función de imitado los caracteres aparecerían con frecuencia (0.5,0.25,0.25) lo cual dista de ser una aproximación estadística razonable. La explicación de este hecho es debido a la utilización de un árbol binario para la representación de los elementos, ya que su distribución estadística siempre será una potencia negativa de 2. Esta potencia depende de la distancia entre la raíz y la hoja correspondiente del árbol.

Existen diferentes mecanismos para disminuir este problema [9]. En general, la existencia de muchos caracteres en el árbol hará que su profundidad sea mayor y la aproximación estadística a la fuente también, así como la utilización de un orden de complejidad mayor.

El límite del orden o del tamaño del texto fuente a procesar viene dado por los recursos hardware involucrados en las operaciones. Por este motivo, en la herramienta Stelin, en el caso de atomicidad basada en palabra, se implementan diferentes modos de actuación con el orden de complejidad, ya que para textos fuentes grandes es posible que el algoritmo desborde en memoria. Por ejemplo, configurar el orden concreto de almacenamiento de términos en cada nivel de recursividad del algoritmo o limitar el orden general de cada nivel a un valor fijo.

El algoritmo implementado, a nivel de palabra en la herramienta Stelin, no tendría por qué cumplir las aproximaciones estadísticas a nivel de carácter justificadas en el algoritmo original de Peter Wayner. Lo cierto es que, por los estudios en curso, la variante implementada se aproxima a la distribución reflejada en la fuente de entrenamiento para órdenes grandes, 8 o más. (véase por ejemplo, Fig.9).

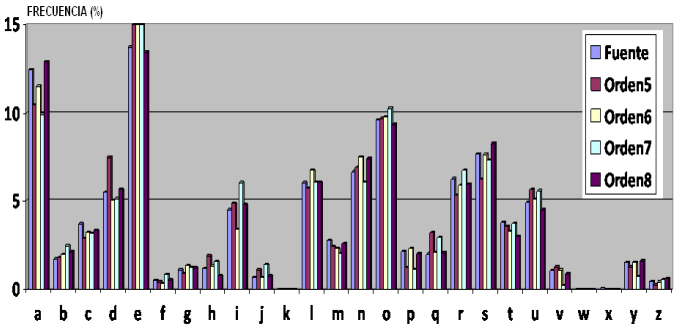


Fig. 9. Ejemplo de comparación de la distribución de frecuencias de caracteres de una fuente de texto de entrenamiento (los 13 primeros capítulos del Quijote con un total de 29.805 Palabras) y un estegotexto generados de ocultar 256 octetos (2048 bits) de información oculta a partir de dicha fuente (como nivel de atomicidad la palabra).

La aproximación estadística a la fuente de entrenamiento a nivel de palabra está por determinar (ver Fig. 10 y 11), ya que es necesario evaluar con más precisión la influencia de diferentes factores relacionados como son: el tipo de fuente de entrenamiento seleccionada, la información a ocultar, el orden de complejidad y el tamaño del estegotexto generado. Los resultados iniciales, a este respecto, no indican que esto simplifique la identificación de un estegotexto distribuido en un canal concreto. No obstante, aparte de otros ataques, está por ver si estas propuestas son seguras frente a ataques basados en localización no equitativa de términos en un estegotexto generado [15]. En este caso concreto, por la información publicada, la propuesta estegoanalítica tendría cierto éxito en análisis sólo si los estegotextos son de al menos de unas cuantos miles de octetos, caso que no se da en los ejemplos de Fig.6 y Fig.7.

Palabra	Fuente	Estego	Palabra	Fuente	Estego
y	3.52%	4.12 %	se	1.10%	1.6%
en	3.61%	4.12 %	un	1.33%	1.37%
el	3.87%	3.89%	mi	0.6%	1.37%
que	3.90%	3.66%	a	1.59%	1.37%
la	3.58%	3.66%	yo	1.21%	1.14%
de	3.91%	2.98%	con	0.74%	1.14%
las	1.4%	1.83%	me	0.74%	0.91%

Fig. 10. Ejemplo de comparación de frecuencias de las palabras más probables en un estegotexto que oculta 16 octetos mediante un orden 10. Texto Fuente: RIMAS (42 KB) de Gustavo Adolfo Bécquer. Factor expansión 1:144.

Palabra	Fuente	Estego	Palabra	Fuente	Estego
que	3.9%	5.6%	yo	1.21%	3.11%
la	3.58%	4.39%	el	3.87%	3.05%
en	3.61%	4.36%	se	1.10%	2.20%
y	3.52%	3.87%	los	1.25%	1.6%
de	3.91%	3.44%	al	1.19%	1.56%
del	1.10%	3.13%	par	0.04%	1.49%

Fig. 11. Ejemplo de comparación de frecuencias de las palabras más probables en un estegotexto que oculta 256 octetos mediante un orden 10. Texto Fuente: RIMAS (42 KB) de Gustavo Adolfo Bécquer. Factor expansión 1:166. Tamaño de estegotexto generado comparable al tamaño de la fuente.

Adicionalmente a los estudios estadísticos, un ataque clásico estegoanalítico a analizar es la posibilidad de un ataque basado en cubierta conocida, es decir, estudiar qué sucedería si un atacante conociera el texto fuente de entrenamiento (que es secreto y compartido entre emisor y receptor) y el orden de complejidad utilizado para generar un estegotexto concreto. Un ataque de este tipo permitiría reconstruir los árboles Huffman correspondiente y decodificar cada palabra del estegotexto a un código concreto. La información recuperada, en general binaria, podría ser analizada posteriormente de diferentes maneras. Un análisis clásico consiste en estudiar su entropía (ecuación de Shannon o aproximación de Shamir y Van Someren [16]) para delatar la existencia de una información cifrada (alta entropía). Este ataque podría ser dificultado de dos formas:

1. La información a ocultar debería ser cifrada, para en el peor de los caos impedir la recuperación de la información original. La herramienta Stelin utiliza un cifrado basado en el algoritmo AES-256 en modo contador, cuya seguridad fue analizada en [17].

2. Los ataques derivados de análisis de entropía y recuperación de información podrían dificultarse de diversas maneras. Actualmente la herramienta Stelin dificulta estos aplicando ideas clásicas de cifradores basados en reducción de redundancia [18]. Stelin utiliza un generador PRNG (AES-256 en modo contador) que asigna a cada rama de cada árbol Huffman una codificación aleatoria (0 o 1) en función de una clave (no de forma fija, por ejemplo, codificación 0 a la rama derecha y 1 a la izquierda), de modo que un atacante que conozca el texto de entrenamiento y el orden de complejidad

tendría dificultades en asignar un código concreto a una palabra del estegotexto determinado. Esto dificultaría, entre otras cosas, extraer la información ocultada y aplicar análisis estadísticos a la misma (por ejemplo, análisis de entropía para revelar la presencia de información cifrada).

5 Conclusiones. Trabajo Futuro.

La esteganografía lingüística es una ciencia que en los últimos años está despertando el interés de la comunidad científica por su enorme potencial. Existen multitud de problemas a solventar, uno de los principales consiste en que el lenguaje natural como medio de ocultación de información es poco redundante (ruidoso) en comparación con otros estegomedios de uso más común (imágenes, vídeo, etc), lo cual dificulta la ocultación de información de forma imperceptible, estadística y lingüísticamente.

La aplicación actual de la esteganografía lingüística en lengua española es muy pobre, de hecho, existen pocos trabajos de interés que analicen algunos de sus aspectos con seriedad. Por este motivo, en el presente artículo se destacan las diferentes líneas de investigaciones actuales en esteganografía lingüística (por ejemplo, la seguridad actual de las CFGs) y se profundiza en un procedimiento de generación automática de estegotextos aplicado a lengua española (propuesta estadística de Peter Wayner). La aplicación directa de esta propuesta (nivel de imitado el carácter) puede producir estegotextos con validez léxica y sintáctica aunque no está exento de errores léxicos, gramaticales o de repetición de términos. Por este motivo, se implementa la herramienta Stelin, disponible en http://stelin.sourceforge.net, para analizar una variante utilizando como nivel de imitado "la palabra". La variante analizada mejora léxica y sintácticamente la generación de estegotextos en lengua española e incluso produce estegotextos con buena apariencia semántica. En la práctica aún con esta mejora resulta realmente complicado ocultar más de unas pocas decenas de octetos, por las pruebas actuales, sin que el estegotexto resultante presente problemas, entre los más destacables, repeticiones de expresiones, semánticos o de coherencia global.

En el artículo se generan estegotextos de fuentes de entrenamiento muy conocidas a modo divulgativo, pero no debe olvidarse que las fuentes de entrenamiento deben ser privadas y a ser posible de difícil acceso para un atacante.

En este sentido, Stelin implementa diferentes mecanismos adicionales para dificultar ataques estadísticos y estegoanalíticos (cifrado de la información y técnicas basadas en reducción de redundancia).

En cualquier caso, las nuevas líneas de investigación abiertas (las que se suponen más prometedoras) analizarán las posibilidades sintácticas y semánticas del lenguaje español para la ocultación de información, así como el análisis de procedimientos robustos de sustitución léxica que puedan abrir nuevos caminos, por separado o unidos con las ideas de generación automática de estegotextos, para crear mecanismos robustos que oculten información de tamaño medio (miles de octetos), en textos en lengua española, con "seguridad" lingüística y esteganográfica.

AGRADECIMIENTOS

Los autores quieren expresar su agradecimiento a la lingüista Irina Mª Argüelles Álvarez por sus consejos y el trabajo en investigaciones paralelas a esta publicación. La doctora Irina trabaja de investigadora en el departamento de lingüística aplicada a la ciencia y tecnología en la Universidad Politécnica de Madrid.

REFERENCIAS

[1] Carracedo, J.: Seguridad en Redes Telemáticas. Mc-Graw Hill InterAmericana de España. ISBN: 84-481-4157-1 (2004), páginas 123-131.

[2] Muñoz, A., Carracedo, J: StegSecret: Una herramienta de estegoanálisis pública**. CIBSI 2007.** Congreso Iberoamericano de Seguridad de la Información. Mar de Plata-Argentina. Noviembre 2007. http://stegsecret.sourceforge.net.

[3] Muñoz, A., Carracedo, J: Herramienta DCST. Automatización de estegoanálisis en Redes Sociales. **X RECSI 2008.** Reunión Española sobre criptología y Seguridad de la información, Salamanca, Sept 2008. España.

[4] Muñoz, A., Carracedo, J., Sánchez, S: Detection of distributed steganographic information in social networks. EATIS 2008. Euro American Conference on Telematics and Information Systems, September 10-12. ACM-DL Proceedings will have ISBN # 978-1-59593-988-3. Copyright © 2008.

[5] Kahn, D: The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet (Hardcover) Scribner 1996. ISBN-13: 978-0684831305.

[6] Chapman, M., Davida, G.I.: Plausible deniability using automated linguistic steganography. In: Davida, G., Frankel, Y. (eds.) InfraSec 2002. LNCS, vol. 2437, Springer, Heidelberg (2002).

[7] Calvo, H., Bolshakov, I.A.: Using selectional preferences for extending a synonymous paraphrasing method in steganography. In: Sossa Azuela, J.H. (ed.) Avances en Ciencias de la Computacion e Ingenieria de Computo -CIC'2004: XIII Congreso Internacional de Computacion, pp. 231–242 (October 2004).

[8] Bergmair, R.: A comprehensive Bibliography of Linguistic Steganography. SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents*, volume 6505, January 2007.*

[9] Wayner,P.: Disappearing Cryptography, Second Edition – Information Hiding: Steganography and Watermarking. Morgan Kaufmann Series; 2 edition (May 2002). ISBN-13: 978-1558607699.

[10] Chapman, M., Davida, G.I.: Hiding the hidden: A software system for concealing ciphertext in innocuous text. In: Han, Y., Quing, S. (eds.) ICICS 1997. LNCS, vol. 1334, pp. 11–14. Springer, Heidelberg (1997)

[11] Lingjun, L., Liusheng, H, Xinxin, Zhao., et al: A statistical attack on Kind of Word-Shift Text-Steganography. IIH-MSP 2008. Pages 1503-1507. 2008. ISBN:978-0-7695-3278-3

[12] Lingyun, X., Xingming, S., Gang, L., Can, G: Research on Steganalysis fort text steganography based on font format. IAS 2007. Page 490-495. 2007. ISBN: 0-7695-2876-7.

[13] Xin-guang, Sui., Hui, Luo., Zhong-liand, Zhu: A steganlaysis Method based on the distribution of Characters. ICSP 2006. 0-7803-9737-1.2006 (IEEE)

[14] Xin-guang, Sui., Hui, Luo., Zhong-liand, Zhu: A steganalysis method based on the distribution of first letters of words. IIH-MSP 2006. 0-7695-2745-0. IEEE.

[15] Zhi-li, Chen., Liu-Sheng, Huang., et al: Effective Linguistic Steganography Detection. IEEE 8th CIT Workshops. 978-0-7695-3242-4. 2008

[16] Shamir, A., Van Someren, N: Playing 'Hide and Seek' with Stored Keys. Lecture Notes in Computer Science. Springer Berlin. Volume 1648/1999. ISBN 978-3-540-66362-1.

[17] Muñoz, A., González, M: PRNG based on new HCI devices entropy sources. Wii ReMote study case. EuroAmerican Conference on Telematics and Information Systems. EATIS PRAGUE 3-5 June 2009.

[18] Hwang, M., A New Redundancy Reducing Cipher. Informatica, vol. 11, no. 4, pp. 435-440, Oct. 2000.

Mecanismo para evitar ataques por confabulación basados en code passing

Marc Jaimez Oscar Esparza Carlos Gañan Javier Parra-Arnau
Universitat Politècnica de Catalunya
{marc.jaimez,oscar.esparza,carlos.ganan,javier.parra }@entel.upc.es

Index Terms—**Mobile agent security, malicious hosts, collusion attack, code passing**

***Resumen*—Los agentes móviles son entidades software formadas por código, datos, itinerario y estado, que pueden migrar de host en host autónomamente ejecutando su código. A pesar de sus ventajas, los aspectos de seguridad restringen enormemente el uso de código móvil. La protección del agente ante ataques de hosts maliciosos, es el problema de seguridad más difícil de resolver en los sistemas de agentes móviles. En particular, los ataques por confabulación han sido poco estudiados en la literatura. Este paper presenta un mecanismo de protección ante ataques por confabulación basados en code passing. Nuestra propuesta es un Multi-Code Agent que contiene diferentes variantes del código para cada host. Una Trusted Third Party es la responsable de proporcionar la información para extraer cada variante, y de tomar referencias temporales que se usarán para verificar la coherencia temporal.**

I. Introducción

Los agentes móviles son entidades software que mueven código y datos a hosts remotos. Los agentes móviles pueden migrar de host en host realizando acciones de forma autónoma o en nombre de un usuario. El uso de agentes móviles permite ahorrar ancho banda, y permite una ejecución autónoma y off-line. Los agentes móviles son especialmente útiles para llevar a cabo tareas automáticamente, en casi todos los servicios electrónicos como el comercio electrónico y la administración de redes. A pesar de sus ventajas, el uso masivo de los agentes móviles se ve restringido por problemas de seguridad [22], [12]. Dentro de este escenario, se consideran dos entidades principales a la hora de estudiar los problemas de seguridad: el agente móvil, y el host que lo ejecuta. Éstos son los principales ataques: (1) el agente ataca al host: la protección del host se consigue usando técnicas de sand-boxing y un apropiado control de acceso [9]; (2) ataque a las comunicaciones: la protección del agente mientras está migrando de host en host se puede asegurar mediante el uso de los protocolos criptográficos conocidos [15]; y (3) el host ataca al agente: no existe por el momento ninguna solución publicada que proteja completamente a los agentes ante estos ataques. Este último ataque es conocido como el problema de los hosts maliciosos.

Este paper introduce un nuevo mecanismo de defensa ante ataques por confabulación basados en code passing. En este tipo de ataque, un host envía el código del agente a otro host que no es el siguiente destinatario de la ruta, y de esta forma este host puede analizar el código del agente antes que llegue a través de la ruta legal. Los ataques por confabulación son difíciles de prevenir o detectar, y es por esta razón que la mayoría de propuestas no los resisten. Solo algunas propuestas intentan limitar estos ataques usando técnicas de protección del itinerario [6], [3], [19]. El uso de nuestro mecanismo implica la ejecución de un agente distinto en cada host. Para hacerlo, generamos distintas variantes del código del agente, y las unimos para construir un *Multi-Code Agent* (MCA), que será la entidad final que se enviará a los hosts. Cuando un hosts reciba el MCA, necesitara cierta información (*extracting instructions*) para poder extraer la variante del agente que le corresponda, estas *extracting instructions* serán solicitadas a una Trusted Third Party (TTP). Éstas peticiones, serán usadas también por la TTP para tomar referencias temporales que servirán para controlar los tiempos de ejecución de los hosts, permitiendo detectar comportamientos maliciosos. El MCA se construye de forma que se pueda evitar que dos o más hosts confabulen para analizar sus variantes y localizar vulnerabilidades en el código.

II. Hosts maliciosos

Los ataques realizados por un host malicioso que está ejecutando un agente móvil son, con diferencia, el problema de seguridad más difícil de resolver en un sistema de agentes móviles [8], [13]. Aunque es posible proteger la migración del agente móvil de escuchas o manipulaciones, mediante el uso de la firma digital o técnicas criptográficas (communications security), es difícil detectar o prevenir ataques realizados por hosts maliciosos durante la ejecución del agente (por ejemplo proporcionar integridad y privacidad). Los hosts maliciosos pueden intentar sacar provecho del agente, si leen o modifican el código, los datos, el flujo de ejecución, las comunicaciones o incluso los resultados. El agente, por supuesto, no puede transportar una clave de descifrado ya que el host podría leerla [5]. Asimismo, si dos o más hosts maliciosos colaboran para realizar un ataque, la tarea de proteger el agente se torna aún más difícil.

En este paper asumimos que los hosts no son entidades de confianza (pueden intentar atacar un agente cuando lo estén ejecutando). Las propuestas publicadas que intentan conseguir integridad de ejecución y privacidad, se pueden dividir en dos categorías principales: propuestas de detección de ataques y propuestas de prevención de ataques. Por un lado, las propuestas de prevención de ataques intentar evitar ataques de eavesdropping o manipulación, antes de que ocurran. Por otro lado, el objetivo de las propuestas de detección de

ataques consiste en prevenir los ataques disuadiendo a los hosts maliciosos.

Obviamente, prevenir es mejor que curar, por lo que las propuestas de prevención son más eficientes en términos de seguridad. Sin embargo, según nuestra opinión las técnicas de prevención de ataques son difíciles de implementar, o sus costes computacionales las hacen difíciles de usar en escenarios reales. De echo, no existe ninguna propuesta de prevención de ataques que proporcione una resistencia satisfactoria. Por otra parte, las técnicas de detección de ataques suelen ser más fáciles de implementar, y por ello las consideramos más prometedoras. No obstante, las técnicas de detección de ataques son efectivas solo en aquellos escenarios en los que exista la posibilidad de penalizar a los hosts maliciosos, y en los que las penalizaciones sean mayores que los beneficios obtenidos por el atacante.

II-A. Propuestas de Prevención de Ataques

Aquí resumimos las principales propuestas que intentan evitar ataques de eavesdropping y manipulaciones, proporcionando privacidad e integridad en la ejecución.

En [23], Ordille propone ejecutar los agentes solo en maquinas de confianza, esto es, máquinas de las cuales no se espera ningún tipo de actividad anómala o maliciosa. Sin embargo, esta propuesta no es útil en una red abierta como Internet porque existen pocos hosts de confianza. Se puede pensar, de forma más general, en un cierto control social y establecer unos varemos de "reputación" de las plataformas que ejecutan los agentes. A partir de estas relaciones de confianza entre entidades es posible inferir otras relaciones que pueden ser usadas para determinar el grado de confianza de un host [14], [20]. Desafortunadamente, estas propuestas no describen los mecanismos de seguridad a utilizar una vez que se ha determinado el nivel de confianza.

Algunos autores proponen el uso de un subsistema cerrado donde se ejecutan de forma segura los agentes y al cual no tiene acceso ni el propio dueño de la plataforma [31], [30], [17]. El principal inconveniente de esta propuesta, es que obligaría a cada host con capacidad para ejecutar agentes a adquirir un equipo hardware. Además, es dudosa la confianza que se puede depositar en el hardware de un determinado suministrador.

En [26], Roth presenta la idea de la protección mutua. En un entorno abierto como Internet, se puede asumir que las relaciones de confianza están limitadas, y por tanto la confabulación entre hosts es poco probable. Por esta razón, el agente móvil se envía junto con otros agentes cooperativos a través de itinerarios disjuntos. El almacenamiento de datos confidenciales y la toma de decisiones del agente móvil se realizan en esos agentes cooperativos, y por tanto los ataques por eavesdropping o manipulaciones no se pueden realizar directamente sobre el agente móvil, sino que deben ser efectuados sobre los agentes cooperativos. Por desgracia, el sistema debe garantizar que las comunicaciones entre los agentes cooperativos sean posibles durante toda la transacción. Adicionalmente, tienen que haber mecanismos para recuperar los resultados en caso que un agente se pierda. Todo y con esto, la posibilidad de confabulación no desaparece por completo.

En [21], se define la entropía de agentes móviles como medida métrica para calcular las intenciones del agente. La idea de esta propuesta es hacer que las intenciones del agente sean desconocidas para el host, porque un host malicioso que no sea capaz de interpretar las intenciones del agente móvil, no será capaz de leer o modificar el agente para sacar beneficio. Esto se puede hacer mediante (1) *intention spreading*, disminuyendola entropía haciendo que el agente ejecute tareas que el usuario no ha pedido; o mediante (2) *intention shrinking*, aumentando la entropía mediante la distribución de las intenciones del usuario en varios agentes cooperativos. En el primer caso, nada evita que los hosts maliciosos obtengan beneficios de todas las tareas. En el segundo caso, tenemos los mismos problemas que en la propuesta previa de protección mutua [26].

La Generación de Claves Dependientes del Entorno presentada en [25] hace que el código del agente sea imposible de descifrar hasta que se den las condiciones propicias, así el análisis previo por parte del host se evita por completo. No obstante, el principal problema de esta propuesta es que el agente es vulnerable una vez ha sido descifrado, y por tanto, la privacidad y la integridad de ejecución no se pueden asegurar. Además, la propuesta obliga a mantener una continua monitorización del entorno.

Una Blackbox es un entorno software que solo permite la lectura de las entradas y las salidas, y los datos internos no pueden ser leídos ni modificados. Desafortunadamente, no hay ningún algoritmo conocido con estas propiedades. La Blackbox limitada en tiempo [10] tiene este nivel de seguridad pero solo durante un periodo de tiempo limitado. Transcurrido este tiempo, ni la privacidad ni la integridad de la ejecución se pueden garantizar. El mecanismo de Hohl se basa en un algoritmo de mess-up que ofusca el código y los datos para hacerlos difíciles de entender, y por ende de modificar. La principal dificultad en este caso es como estimar el tiempo durante el cual la ejecución es segura. Por otrolado, tampoco existen mecanismos para evaluar la bondad de un algoritmo de mess-up. Adicionalmente, un host malicioso con suficiente tiempo y recursos podría analizar el agente ofuscado para sacar información, o manipular el agente para obtener una ejecución favorable.

El uso de programas cifrados [27] se propone como la única manera de proporcionar privacidad y integridad de ejecución al código móvil. Los hosts ejecutan el código cifrado directamente. La función para descifrar, es utilizada para recuperar los resultados cuando el agente llega al host origen. En [4], la propuesta se mejora de forma que los agentes pueden atravesar múltiples hosts. En [1], el esquema propuesto permite a los agentes tomar decisiones mientras viajan, gracias al uso de una TTP. La dificultad con la que se encuentran estas propuestas, es encontrar funciones que pueden ser ejecutadas de forma cifrada.

II-B. Propuestas de Detección de Ataques

Aquí resumimos las principales propuestas publicadas que intentan detectar manipulaciones para proporcionar integridad de ejecución.

En [18], se introduce la idea de replicación y voto. En cada etapa, los hosts ejecutan el agente en paralelo y envían varias réplicas del agente a un conjunto independiente de hosts en la siguiente etapa. En algunas de las etapas, los hosts comparan los resultados de los agentes y escogen los resultados correctos por mayoría. Esta propuesta no solo proporciona un mecanismo de tolerancia a fallos, sino que detecta aquellos hosts que han realizado un ataque de manipulación. Se asume que los resultados de todas las réplicas deben ser los mismos si todos los hosts han actuado honestamente, por tanto todos los hosts en una misma etapa tienen que tener los mismos recursos y datos. Desafortunadamente, esto no es coherente con la propiedad de independencia de los host, por ejemplo los hosts pueden tener distintos intereses para atacar al agente.

En [29], [28], Vigna introduce la idea de las trazas criptográficas, que son logs de las operaciones realizadas por el agente durante su ejecución. Con dichas trazas, se puede volver a ejecutar el agente para verificar su ejecución en el host. Si el host origen sospecha que un host ha modificado el agente y quiere verificarlo, pide las trazas y vuelve a ejecutar el agente. Si la nueva ejecución no coincide con las trazas, el host nos esta intentando engañar. En lugar de las trazas, el host envía un hash de estas para evitar ataques de repudio. La propuesta no solo detecta manipulaciones, sino que también proporciona una prueba del comportamiento malicioso del host. Sin embargo, esta propuesta tiene dos inconvenientes: (1) la verificación solo se lleva a cabo en caso de sospecha, pero no se explica como se detecta a un host sospechoso; (2) durante un periodo indeterminado, cada host tiene que reservar suficiente capacidad para almacenar las trazas ya que el host origen se las puede pedir. Estos inconvenientes pueden ser aliviados mediante el control del tiempo de ejecución del agente en los hosts [7], aun así, consideramos que el uso de trazas sigue siendo muy costoso para todas las entidades involucradas.

En la propuesta de los Estados de Referencia[1][11], la correcta ejecución en un host se verifica en el siguiente host. Esta propuesta se basa en las trazas criptográficas de Vigna, pero tiene el problema de enviar los datos de entrada del agente (que pueden ser confidenciales) al siguiente host para verificar la ejecución. Algunas propuestas posteriores están basados también en las trazas de Vigna [32], [16], pero ninguna de ellas soluciona el problema de detectar manipulaciones de forma satisfactoria.

En [2] se usa un agente clon para realizar una ejecución de referencia del agente. Si un host ha alterado el código del agente o la ejecución, podemos compararla con la obtenida por el clon de referencia. Con esta propuesta podemos detectar los ataques prácticamente en el momento de llevarse a cabo. Sin embargo, las continuas verificaciones de la ejecución implican una constante comunicación entre agentes. Además, la ejecución de dos agentes (uno ejecutado por el host, y el otro en una TTP) conlleva un gasto mayor de ancho de banda y recursos computacionales. En [24], distintos agentes (agentes móviles y agentes sedentarios) cooperan, el agente móvil es ejecutado por los hosts y los agentes sedentarios son ejecutados en plataformas de confianza que generan ejecuciones de referencia.

[1]Los estados de referencia son aquellos que han sido producidos en hosts de referencia (de confianza)

III. Multi-Code Agent

En este paper presentamos un nuevo mecanismo que permite resistir ataques por confabulación basados en Code Passing. Tal y como hemos mencionado previamente, los ataques por confabulación en sistemas de agentes móviles apenas han sido estudiados. En un ataque basado en Code Passing, un host envía el código del agente a otro host que no es el siguiente destinatario en la ruta del agente, sino que es cualquier otro dentro del itinerario; gracias a esto, este host puede disponer del código del agente mucho antes de recibirlo a través de la ruta legal, y por tanto puede aprovechar ese tiempo para analizarlo.

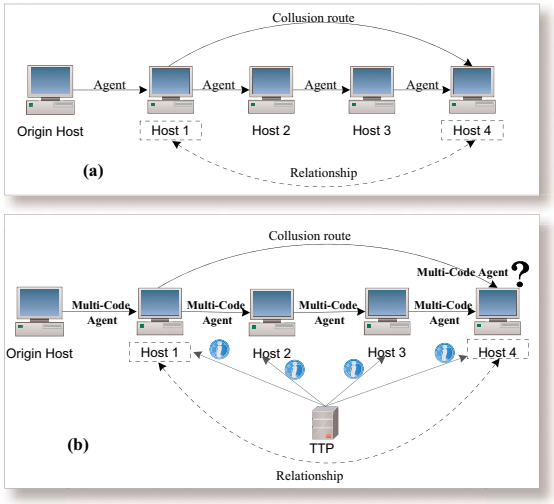


Figura 1. El problema del Code Passing

En la Figura 1.a se muestra un ataque por confabulación basado en Code Passing: el host origen envía el agente móvil al Host-1, que ejecuta el agente de forma normal y lo envía al próximo host (Host-2) a través del itinerario marcado. Al mismo tiempo, el Host-1 envía el código del agente a otro host dentro del itinerario, llamémosle Host-4, quien dispondrá de más tiempo para analizar el código del agente en busca de posibles vulnerabilidades.

Para evitarlo, generaremos diferentes *variantes* del código, las uniremos, y construiremos un Multi-Code Agent (MCA), que será la entidad final que se enviará a todos los hosts. Cuando un hosts reciba el MCA, necesitara cierta información (*extracting instructions*) para poder extraer la variante del agente que le corresponda, estas *extracting instructions* serán solicitadas a una Trusted Third Party (TTP). A su vez, la TTP se encargará de obtener referencias temporales para controlar los tiempos de ejecución de los host. Finalmente, cuando el host finalice la ejecución de su variante, enviará elMCA al siguiente host y el proceso empezará de nuevo. El MCA se diseña por tanto con el objetivo de evitar que dos o mas hosts confabulen y puedan analizar el código de sus variantes para

hallar vulnerabilidades. Las referencias temporales se usan para limitar el tiempo de ejecución en cada host, con lo que se pueden detectar comportamientos maliciosos.

En la Figura 1.b se muestra la idea principal de este mecanismo. El host origen envía el MCA al primer host del itinerario, Host-1. Este solicita las *extracting instructions* a la TTP para poder extraer su correspondiente variante, y a su vez la TTP usa esta solicitud para tomar una referencia temporal de confianza. Seguidamente, el Host-1 ejecuta su variante y envía el MCA al siguiente host en el itinerario, Host-2. El Host-1 puede enviar el MCA (o incluso su propia variante) a otro host directamente, por ejemplo al Host-4, a través de la ruta de confabulación. Sin embargo, este último no podrá extraer su variante sin disponer de las *extracting instructions* pertinentes. Como la TTP controla los tiempos de ejecución, el Host-4 no podrá solicitar las *extracting instructions* hasta que el Host-3 le envíe el MCA.

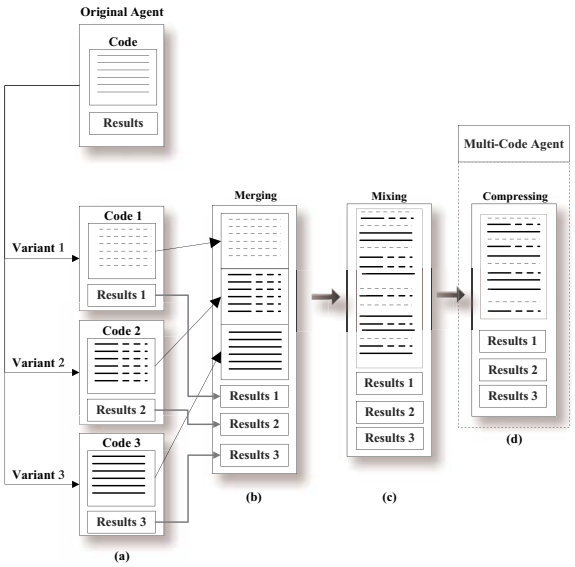


Figura 2. Etapa de codificación

III-A. Etapa de Codificación

La etapa de codificación empieza con la modificación del agente original para obtener las diferentes variantes (ver Figura 2.a). Tal como se muestra, el agente original tiene una sección de Código, que contiene todo el Bytecode del agente, y una sección de Resultados que hace referencia a la estructura de datos dónde se guardan los resultados de la ejecución. En este ejemplo en particular hay tres variantes del agente original, ya que el agente se va a ejecutar en tres hosts distintos. Estas nuevas variantes del agente deben cumplir dos funciones básicas: proporcionar Bytecodes distintos, y proporcionar una estructura de datos distinta para los resultados de ejecución de cada host. Después de obtener todas las variantes, el codificador toma el Bytecode de cada una de ellas y construye el MCA. Este proceso se divide en tres fases: *Merging*, *Meshing up*, y *Compressing*. En la fase de *merging* (ver Figura 2.b) se toma el Bytecode de todas las variante y se pone todo junto en un único archivo. En la fase de *meshing up* (ver Figura 2.c) se mezclan las instrucciones Bytecode de los distintas variantes. Finalmente, en la fase de *compressing* (ver Figura2.d) se elimina redundancia de código. Al final de la fase de codificación, tenemos un MCA que tiene una sección de Código (que contiene el Bytecode de todas las variantes) y una sección de Resultados dónde se almacenan los resultados de cada host.

III-A1. Creación de los nuevos agentes: El Multi-Code Agent contiene el código de todas las variantes del agente original, y estas deben cumplir dos funciones básicas: proporcionar distintos Bytecodes (queremos que cada host ejecute un agente distinto), y proporcionar distintas estructuras de datos para los resultados de ejecución (queremos verificar que cada host ha ejecutado el agente correcto). El proceso de creación de las variantes del agente original se divide en dos pasos:

III-A1a. Paso1. Modificación del código para obtener diferentes estructuras de datos: En este paso, modificamos el código del agente original para que cada una de las variantes genere una sección de Resultados distinta. Cada una de estas secciones de Resultados contendrá una *Identity Mark*, que servirá para comprobar que cada host ha ejecutado su correspondiente variante. La forma en que cada host dispondrá toda esta información (los propios valores, su orden y sus relaciones) en la sección de Resultados definirá como implementa su *Identity Mark*. La sección de Resultados, finalmente se adjuntará al MCA para ser enviado al siguiente host.

III-A1b. Paso2. Modificación del código para obtener variabilidad en el Bytecode: El objetivo de este paso es obtener un conjunto de nuevas variantes, distintas y difíciles de analizar, que realicen las mismas tareas que el agente original pero usando un código distinto. Sin embargo, recomendamos que estas nuevas variantes tengan una estructura similar (mismas clases, métodos, nombres, variables, etc) y solo se modifique el código de los métodos para introducir diferencias entre variantes. Si respetamos esta restricción, seremos capaces de reducir significativamente la longitud del MCA y la longitud de las *extracting instructions*. Es importante recalcar que cuando decimos que todas las variantes deben tener una estructura similar, esto no significa que se deba mantener la estructura del agente original.

Para ilustrar la importancia de mantener la estructura básica de las clase en todas las variantes, mostraremos un ejemplo con un agente original y dos variantes. La Figura 3.a muestra el código del agente original, que esta compuesto por una única clase llamada OriginalAgentCode. Esta clase contiene un único método llamado method1(), que realiza un simple cálculo aritmético $a = 10 + 20$, y devuelve el resultado. Como se puede ver, para crear las nuevas variantes mantenemos la misma estructura del agente original (una clase y un método) pero añadimos dos nuevas variables, b y c. Estas nuevas variables nos permitirán realizar las mismas operaciones de manera distinta. La Figura 3.b muestra el código de la primera variante, y la Figura 3.c muestra el código de la segunda variante.

Si analizamos el código del ejemplo, ambas variantes tienen una longitud de 3328 bits, pero tan solo 144 bits de los 3328 son distintos para cada variante. Esto significa que solo

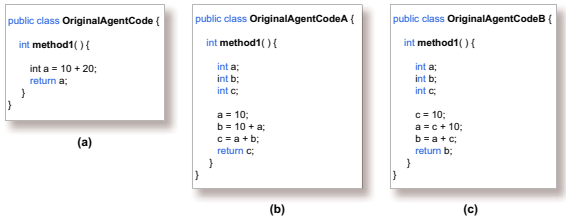


Figura 3. Código original y código de las nuevas variantes

un 3,42 % del Bytecode difiere de una variante a otra, y es debido al hecho que hemos usado la misma estructura para las dos variantes. Si hubieran más de dos hosts en el itinerario, necesitaríamos mas variantes del agente original, y por tanto deberíamos implementar la misma operación aritmética de manera distinta para cada nueva variante.

III-A2. Construyendo una Identity Mark: La *Identity Mark* es un vector que contiene n valores. Estos valores pueden ser resultados de ejecución R_i, o pueden ser valores intermedios V_j. Un ejemplo sencillo de una posible implementación de una *Identity Mark* con $n = 5$ se muestra en la Figura 4. El Host ejecuta su variante y obtiene dos valores que se corresponden a los resultados de ejecución (R_1y R_2), y tres valores intermedios (V_1, V_2, V_3). Con todos estos valores, el agente construye una estructura de datos que implementa su *Identity Mark*, y finalmente se envía junto al MCA.

La fortaleza de la *Identity Mark* se basa en dos características principales:

1. El orden de los valores: la *Identity Mark* es un vector ordenado de n valores, por tanto, existen $n!$ *Identity Marks* distintas que se pueden obtener simplemente ordenando los valores de manera distinta. De esta manera, la probabilidad de generar la *Identity Mark* de una cierta variante es de $\frac{1}{n!}$. En el ejemplo de la Figura 4, la *Identity Mark* contiene 5 valores (R_1, R_2, V_1, V_2, V_3), que pueden generar ciento veinte secuencias distintas.
2. Los valores intermedios: no solo el orden de los valores es específico de cada agente, sino que los propios valores se obtienen de forma distinta dependiendo de la variante. Nosotros asumimos que los valores correspondientes a los resultados de ejecución (R_i) no pueden ser cambiados. Sin embargo, los valores intermedios V_j se pueden calcular de distintas formas. Por ejemplo, en el ejemplo de la Figura 4, el cálculo de V_2 en el Agente A podría depender de R_1, V_1 o cualquier otro dato de entrada usado solo por el Agente A. Por otro lado, el cálculo de V_2 en el Agente B puede depender solo de R_2. Gracias a esto, un host no puede establecer ninguna relación directa entre el V_2 del Agente A, y el V_2 del Agente B.

Tal y como se puede deducir, si incrementamos el número de valores n del vector *Identity Mark,* aumentaremos las posibilidades de proteger el agente de ataques por confabulación.

III-A3. Merging: En esta fase, compilamos los archivos .java, obtenemos los archivos .class, y los unimos para construir el MCA. Siguiendo con el ejemplo propuesto en Subsección III-A1, compilamos los archivos OriginalAgentCodeA.java y

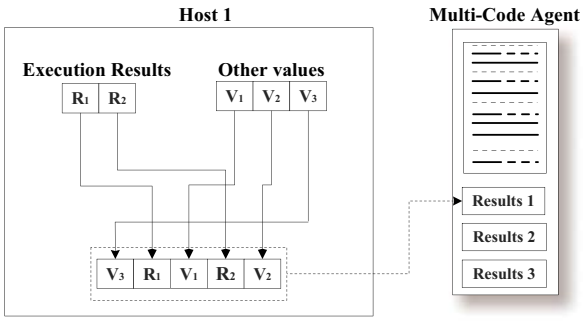


Figura 4. Ejemplo de una Identity Mark

OriginalAgentCodeB.java, y obtenemos dos nuevos archivos: OriginalAgentCodeA.class y OriginalAgentCodeB.class.

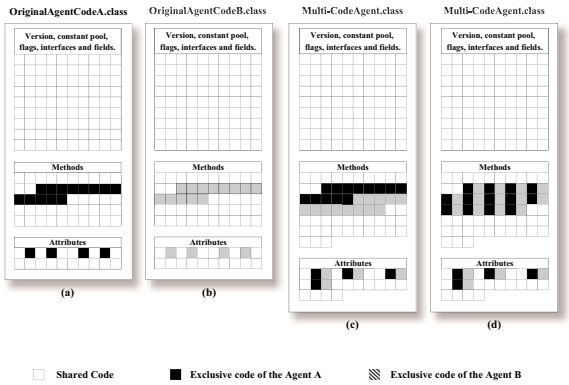


Figura 5. Building up the Multi-Code agent

La Figura 5.a y la Figura 5.b nos muestran sendos diagramas representativo de la estructura de los archivos .class. Las celdas blancas representan el código compartido por ambas variantes, las celdas negras representan el código exclusivo de la variante A, y las celdas grises representan el código exclusivo de la variante B. La Figura 5.c muestra como quedaría la estructura del archivo .class del Multi-Code Agent. Como podemos observar, hemos construido un único archivo .class que contiene dos implementaciones distintas del agente original.

Un ejemplo detallado de la fase de Merging se da en la Figura 6. Partiendo del agente original (ver Figura 6.a), se crean dos nuevas variantes. La Figura 6.b se corresponde con el Bytecode específico del method1() de la variante A (color negro). La Figura 6.c se corresponde con el Bytecode específico del method1() de la variante B (color negro). Finalmente, la Figura 6.d muestra el Bytecode resultante del MCA (sin haber pasado todavía por las fases de mixing y compressing). Por simplicidad, el resto del Bytecode no se muestra en la Figura 6.

III-A4. Mixing: Hemos partido de un agente original y lo hemos modificado para obtener dos implementaciones distintas del mismo. Seguidamente, estos dos nuevos agente han sido unidos para generar un MCA, y ahora vamos a mezclar su código. La idea básica de la fase de Mixing se muestra en

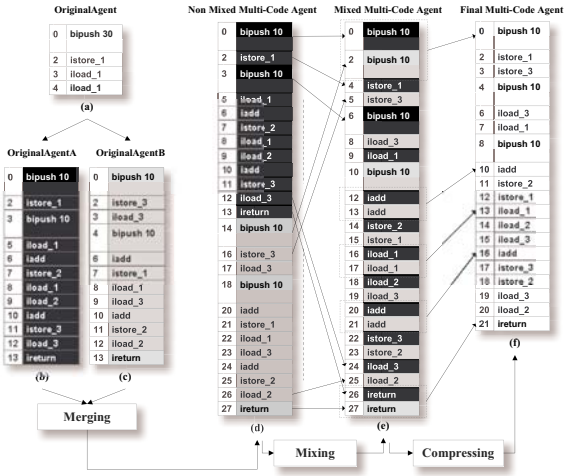


Figura 6. A bytecode view of the Multi-Code Agent construction

la Figura 5.d. Antes de realizar el Mixing, las celdas negras y grises están separadas y por tanto podrían ser identificadas por un atacante. Después del Mixing, un atacante tendrá más dificultades para identificar las celdas de cada variante.

Un ejemplo detallado de la fase de Mixing se puede ver en la Figura 6.e, que nos muestra una posible distribución final de las instrucciones Bytecode después del Mixing. Es importante recalcar que el orden en que disponemos las instrucciones dentro del MCA puede ser cualquiera. De echo, cuanto más alteremos dicho orden, más difícil y complicado será entender el código. Sin embargo, hay que tener en cuenta que la complejidad y la longitud de las *extracting instructions* también se verán incrementadas. En nuestro caso, preservaremos el orden natural de las instrucciones para simplificarlas .

III-A5. Compressing: La última fase dentro de la etapa de Codificación del MCA consiste en comprimir el conjunto de instrucciones Bytecode. Como resultado de la fase de Mixing, algunas instrucciones consecutivas son idénticas, y cada vez que se de este echo podemos eliminar una de ellas (ver Figura 6.f). Aunque la longitud del código se reduzca al eliminar algunas instrucciones, el factor de compresión es muy bajo. En el caso del ejemplo, la longitud del MCA antes de la compresión es de 3472 bits, y la longitud después de la compresión es de 3424 bits. Esto significa que hemos reducido la longitud del MCA en un 1.38 %, que es prácticamente despreciable. Todo y con esto, la fase de compresión es necesaria para añadir complejidad al código resultante. Para un atacante resulta más difícil extraer una variante si algunas instrucciones han sido borradas.

III-A6. Extracting instructions: Cuando un host recibe el MCA, debe extraer su correspondiente variante, y para hacerlo necesita saber que instrucciones pertenecen a su propio agente. Esta información está contenida en lo que llamamos *extracting instructions*, que es sencillamente una lista de posiciones de memoria. Estas posiciones de memoria indican que instrucciones Bytecode no pertenecen a nuestra variante (lista negra). La Figura 7 nos muestra el proceso de generación de las *extracting instructions*. El MCA contiene una mezcla comprimida del código de las distintas variantes (ver Figura 7.a). Las celdas negras contienen instrucciones del agente A; las celdas blancas contienen instrucciones del agente B; y las celdas blancas contienen instrucciones compartidas por ambos agentes. Por ejemplo, para obtener el agente A, solo necesitamos preservar aquellas celdas que contienen instrucciones del agente A (ver Figura 7.b). Lo mismo ocurre para extraer el agente B (ver Figura 7.c). Una vez tenemos identificadas las celdas que contienen las instrucciones de otros agentes, guardamos esta información en un vector (ver Figura 7.d, y Figura 7.e) y lo mandamos a la TTP .

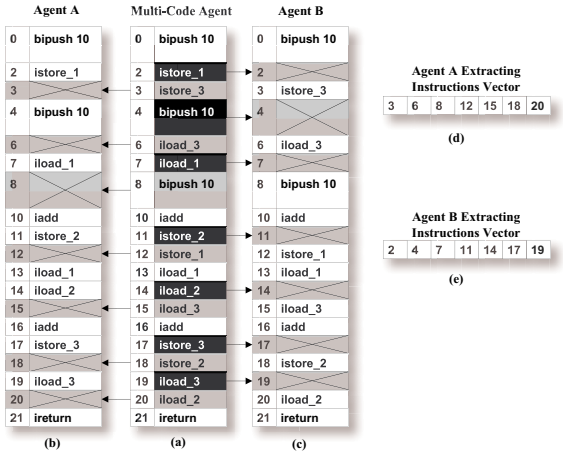


Figura 7. Generación de las Extracting Instructions

III-B. Etapa de Distribución

Antes de enviar el MCA, el host origen tiene que enviar todas las *extracting instructions* a la TTP (un vector por cada host del itinerario). Ésta sen encargará de gestionar el envío de las *extracting instructions* a los hosts. Gracias a esto, el host origen queda liberado de esta tarea y no tiene que permanecer online mientras el MCA realiza su migración. Este proceso se muestra en la Figura 8:

- Paso 1: el host origen envía el conjunto de *extracting instructions* (I_1, I_2,..., I_N) a la TTP para su distribución (en el ejemplo $N = 2$).
- Paso 2: el host origen envía el MCA al primer host del itinerario (Host-1). Después de este paso, el host origen puede estar offline hasta que el MCA salga del último host con todos los resultados.
- Paso 3: el Host-1 recibe el MCA y realiza una petición de *extracting instructions* a la TTP.
- Paso 4: la TTP autentica la petición del Host-1, y le envía su *extracing instructions vector*, I_1. Además almacena una referencia temporal T_1 del instante en el que el Host-1 ha pedido el vector.
- Paso 5: el Host-1 extrae y ejecuta su variante, y adjunta los resultados de ejecución D_1 al MCA antes de enviarlo al siguiente host (Host-2).
- Paso 6 y 7: el resto de hosts del itinerario realizan el mismo proceso que el Host-1.

- Paso 8: cuando el último host ha recibido sus *extracting instructions*, la TTP envía todas las referencias temporales recogidas (T_1, ..., T_N) al host origen .
- Paso 9: el último host envía el MCA con todos los resultados de ejecución (D_1, ... ,D_N) al host origen.

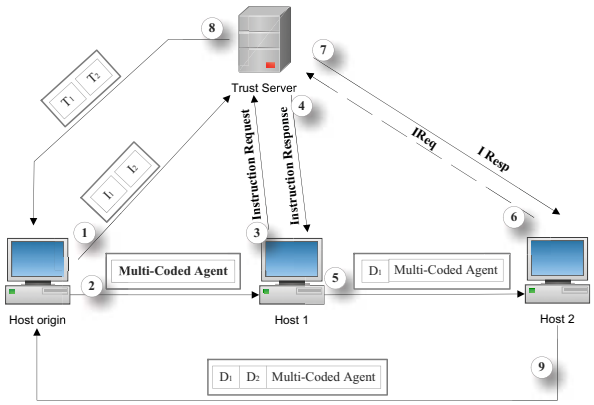


Figura 8. Distribucion del Multi-Code Agent

III-C. Etapa de verificación de las referencias temporales

La TTP ha estado tomando referencias temporales cada vez que ha recibido una petición de *extracting Instructions* (IReq) válida. Gracias a esto, es posible calcular el periodo de tiempo en el que cada host ha dispuesto de su variante. La Figura 9 nos muestra un diagrama temporal en el que se puede apreciar que la TTP toma la referencia temporal justo en el momento en que envía el Instruction Response message (IResp) al host. Estas referencias temporales se utilizan en el host origen para verificar la coherencia de tiempos de ejecución en los hosts. El host origen calcula $\triangle T_i = T_{i+1} - T_i$, que incluye el tiempo de extracción del agente, el tiempo de ejecución de agente, el tiempo de transmisión del MCA al siguiente host, el retardo de propagación, y el intervalo de tiempo desde que el host siguiente recibe el MCA hasta que este recibe el IResp. Con toda esta información, el host origen es capaz de estimar el tiempo de ejecución de cada agente.

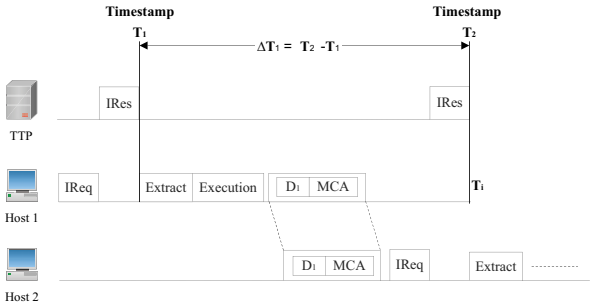


Figura 9. Timestamps storing

Dependiendo de las características computacionales de los hosts, y de los recursos necesarios para ejecutar los agentes, el host origen estima un tiempo máximo de ejecución permitido

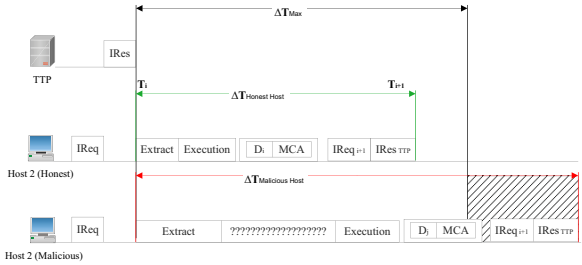


Figura 10. Ejecución honesta vs ejecución maliciosa

($\triangle T_{Max}$). En el caso que algún $\triangle T_i$ sea mayor que este valor, el host será etiquetado como sospechoso de realizar actividades maliciosas. La Figura 10 muestra dos lineas temporales, una en la que el Host-2 actúa honestamente, y otra en la que actúa maliciosamente. Si el Host-2 actúa honestamente, el host origen detectará que $\triangle T_{HonestHost} < \triangle T_{Max}$. Por otro lado, si detecta que $\triangle T_{MaliciousHost} > \triangle T_{Max}$, entonces el Host-2 será sospechoso de actuar maliciosamente.

III-D. Fase de verificación de la coherencia de los datos

Como hemos mencionado con anterioridad, cada agente tiene una Identity Mark propia, que se codifica en los resultados de ejecución. Analizando estas Identity Marks, seremos capaces de verificar que variante ha sido ejecutada por cada host. Concretamente, al final del proceso de distribución, el host origen recibe el MCA con los resultados de ejecución, y una por una va extrayendo todas las Identity Marks. Si estas corresponden al host que las ha creado, entonces tenemos un resultado positivo y asumimos que este host en particular ha ejecutado su variante asignada. En caso contrario, asumimos que el host ha ejecutado cualquier otra variante, y por lo tanto lo etiquetamos como sospecho.

IV. Conclusiones

Este paper introduce un nuevo mecanismo para evitar ataques por confabulación basados en Code Passing. El mecanismo se basa en construir un Multi-Code Agent que contiene diferentes variantes del código de un agente original. Cada variante se ejecuta en un host distinto, y gracias a esto, cualquier información compartida por los hosts (en una confabulación) no sirve para obtener ninguna ventaja. La introducción de una TTP, que se encarga de distribuir las extracting instructions y tomar referencias temporales, nos permite detectar y evitar ataques por Code Passing. Aunque el Multi-Code Agent contiene diferentes variantes del código de un mismo agente, la longitud de este es prácticamente igual a la de un único agente. Por tanto, evitamos malgastar ancho de banda, que es un inconveniente típico de las propuestas basadas en el uso de varios agentes. Con la inclusión de una Identity Mark en los resultados de ejecución del agente, podemos verificar que cada host ha ejecutado su correspondiente variante. Finalmente, el uso de una TTP para obtener referencias temporales de confianza, nos permite limitar el tiempo de ejecución utilizado por los hosts.

Acknowledgements

This work was supported the Spanish Government through projects CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES", TSI2007-65393-C02-02 "ITACA" and TSI2005-07293-C02-01 "SECONNET", and by the Government of Catalonia under grant 2005 SGR 01015 to consolidated research groups.

REFERENCIAS

[1] J. Algesheimer, C. Cachin, J. Camenisch, and G. Karjoth. Cryptographic security for mobile code. In *IEEE Symposium on Security and Privacy*, 2001.

[2] L. Benachenhou and S. Pierre. Protection of a mobile agent with a reference clone. *Computer Communications*, 29(2):268–278, 2006.

[3] J. Borrell, S. Robles, J. Serra, and A. Riera. Securing the Itinerary of Mobile Agents through a Non-Repudiation Protocol. In *IEEE International Carnahan Conference on Security Technology*, 1999.

[4] C. Cachin, J. Camenisch, J. Kilian, and Joy Müller. One-round secure computation and secure autonomous mobile agents. In *27th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 1853 of *LNCS*. Springer-Verlag, 2000.

[5] D. Chess. Security considerations in agent-based systems. In *First IEEE Conference on Emerging Technologies and Applications in Communications (etaCOM)*, 1996.

[6] C. Unger F. Kaderali D. Westhoff, M. Schneider. Methods for Protecting a Mobile Agent's Route. In *ISW'99*, volume 1729 of *LNCS*. Springer-Verlag, 1999.

[7] O. Esparza, J.L. Muñoz, M. Soriano, and J. Forné. Punishing Malicious Hosts with the Cryptographic Traces Approach. *New Generation Computing*, 24(4):351–376, 2006.

[8] W.M. Farmer, J.D. Guttman, and V. Swarup. Security for mobile agents: issues and requirements. In *19th National Information Systems Security Conference*, 1996.

[9] S. Haridi, P. Van Roy, P. Brand, and C. Schulte. Programming languages for distributed applications. *New Generation Computing*, 16(3):223–261, 1998.

[10] F. Hohl. Time Limited Blackbox Security: Protecting Mobile Agents From Malicious Hosts. In *Mobile Agents and Security*, volume 1419 of *LNCS*. Springer-Verlag, 1998.

[11] F. Hohl. A Framework to Protect Malicious Hosts Attacks by Using Reference States. In *International Conference on Distributed Computing Systems (ICDCS)*, 2000.

[12] W. Jansen. Countermeasures for Mobile Agent Security. *Computer Communications, Special Issue on Advanced Security Techniques for Network Protection*, 2000.

[13] W. Jansen and T. Karygiannis. Mobile Agent Security. Special publication 800-19, National Institute of Standards and Technology (NIST), 1999.

[14] H. Kim and L. Moreau. Trust Relationships in a Mobile Agent System. In *5th International Conference on Mobile Agents (MA'2001)*, volume 2240 of *LNCS*. Springer-Verlag, 2001.

[15] D. Kinny. Reliable agent communication - a pragmatic perspective. *New Generation Comput.*, 19(2):139–156, 2001.

[16] K.K. Leung and K.W. Ng. Detection Of Malicious Host Attacks by Tracing with Randomly Selected Hosts. In *International Conference on Embedded And Ubiquitous Computing*, volume 3207 of *LNCS*. Springer-Verlag, 2004.

[17] A. Maña, J. Lopez, J.J. Ortega, E. Pimentel, and J.M. Troya. A framework for secure execution of software. *International Journal of Information Security*, 3(2):99–112, 2004.

[18] Y. Minsky, R. van Renesse, F. Schneider, and S.D. Stoller. Cryptographic Support for Fault-Tolerant Distributed Computing. In *Seventh ACM SIGOPS European Workshop*, 1996.

[19] J. Mir and J. Borrell. Protecting Mobile Agent Itineraries. In *Mobile Agents for Telecommunication Applications (MATA 2003)*, volume 2881 of *LNCS*. Springer-Verlag, 2003.

[20] G. Navarro, S. Robles, and J. Borrell. Role-Based Access Control for E-commerce Sea-of-Data Applications. In *Information Security Conference (ISC'02)*, volume 2433 of *LNCS*. Springer-Verlag, 2002.

[21] S.K. Ng. *Protecting Mobile Agents Against Malicious Hosts*. PhD thesis, The Chinese University of Hong Kong, 2002.

[22] R. Oppliger. Security issues related to mobile code and agent-based systems. *Computer Communications*, 22(12):1165–1170, 1999.

[23] J. Ordille. When agents roam, who can you trust? Technical report, Computing Science Research Center, Bell Labs, 1996.

[24] A. Ouardani, S. Pierre, and H. Boucheneb. A security protocol for mobile agents based upon the cooperation of sedentary agents. *J. Network and Computer Applications*, 30(3):1228–1243, 2007.

[25] J. Riordan and B. Schneier. Environmental Key Generation Towards Clueless Agents. In *Mobile Agents and Security*, volume 1419 of *LNCS*. Springer-Verlag, 1998.

[26] V. Roth. Mutual protection of cooperating agents. In *Secure Internet Programming: Security Issues for Mobile and Distributed Objects*, volume 1906 of *LNCS*. Springer-Verlag, 1999.

[27] T. Sander and C.F. Tschudin. Protecting mobile agents against malicious hosts. In *Mobile Agents and Security*, volume 1419 of *LNCS*. Springer-Verlag, 1998.

[28] G. Vigna. Protecting Mobile Agents through Tracing. In *Proceedings of the Third International Workshop on Mobile Object Systems*, 1997.

[29] G. Vigna. Cryptographic traces for mobile agents. In *Mobile Agents and Security*, volume 1419 of *LNCS*. Springer-Verlag, 1998.

[30] U. G. Wilhelm, S. Staamann, and L. Buttyán. Introducing trusted third parties to the mobile agent paradigm. In *Secure Internet Programming: Security Issues for Mobile and Distributed Objects*, volume 1603 of *LNCS*. Springer-Verlag, 1999.

[31] B.S. Yee. A sanctuary for mobile agents. In *DARPA workshop on foundations for secure mobile code*, 1997.

[32] C.M. Yu and K.W. Ng. A flexible tamper-detection protocol for mobile agents on open networks. In *International Conference of Information and Knowledge Engineering(IKE'02)*, 2002.

Extendiendo TLS para el soporte de transmisión multicanal seguro en señalización

Daniel Díaz-Sánchez, Fabio Sanvido, Andrés Marín-López, Florina Almenárez-Mendoza, Alberto Cortés-Martín
Departamento de ingeniería telemática,
Universidad Carlos III de Madrid
Avda de la Universidad 30, E-28911, Leganés (Spain)
{dds,dproserp,amarin,florina,alcortes}@it.uc3m.es.

***Resumen*—SCTP es un protocolo de nivel de transporte, orientado a conexión, capaz de proporcionar servicios avanzados de multihoming y multistreaming. Gracias a sus características de fiabilidad y redundancia se propone como la elección más oportuna para transporte de señalización, sobre todo en los entornos de telefonía sobre IP. Por otro lado, otra característica interesante de SCTP es la protección contra el head-of-line que permite disponer de varios streams para multiplexar el tráfico de aplicación evitando bloqueos. Sin embargo, cuando TLS se utiliza con SCTP para proteger los flujos de datos presenta un problema de cuello de botella. Este artículo propone una solución para el uso de TLS sobre SCTP que permite mantener las propiedades del protocolo SCTP a la vez que mejora el soporte de seguridad utilizando capas internas de TLS. De esta manera se mejoran las prestaciones y el nivel de seguridad actual. El artículo presenta además resultados experimentales que demuestran claramente un incremento de rendimiento muy alto en comparación con la propuesta de TLS sobre SCTP definida en la RFC 3436.**

***Index Terms*—SCTP, TLS, transport, security**

I. Introducción

SCTP es un protocolo de nivel de transporte fiable y orientado a conexión. Combina características propias tanto de TCP como de UDP y además varias funcionalidades añadidas [1], [2]. SCTP fue diseñado principalmente para superar las limitaciones de TCP para su uso como transporte de protocolos de telefonía sobre IP (Sistema de Señalización 7). Hoy en día también es usado para transportar otros protocolos, por ejemplo Diameter. SCTP soporta multihoming: los extremos de una conexión SCTP pueden tener múltiples direcciones IP, que añaden redundancia en las rutas de red. La fiabilidad que el multihoming permite es uno de los temas clave en el transporte de SS7 sobre IP. Además SCTP separa las funciones de fiabilidad de los datos de las funciones de ordenamiento de los mensajes. Esto habilita múltiples subflujos ordenados en una sola conexión fiable y provee una solución al problema del "head-of-line" blocking presente en otros protocolos como TCP.

Los streams SCTP son canales lógicos unidireccionales entre los extremos de la asociación SCTP a través de los cuales los mensajes del usuario se entregan en secuencia (exceptuado los enviados con el servicio de entrega no ordenada). Un paquete SCTP está formado por uno o más *chuncks* los cuales representan la unidad de información fundamental de SCTP y están constituidos por una parte de cabecera y una parte de datos. Empleando este concepto, partes de varios mensajes de usuario pueden ser multiplexadas en un mismo paquete donde cada *chunck* es usado para transportar un trozo de mensaje dentro del paquete SCTP. Los paquetes incluyen una cabecera común, posibles *chunks* de control y los *chunks* de datos que encapsulan los datos del usuario.

SCTP puede ser utilizado en diferentes escenarios. En un contexto de telefonía, SCTP transporta diferentes conversaciones y la señalización necesaria en diferentes stream. SCTP también permite su uso en protocolos de aplicación como HTTP para evitar el head-of-line blocking, donde diferentes recursos son enviados por diferentes stream y descargados independientemente (véase el trabajo del Protocol Engineering Laboratory de la University of Delaware in HTTP over SCTP Multistreaming o Multiple File Transfer using SCTP Multistreaming).

II. Soporte de seguridad con Transport Layer Security

A fin de proporcionar seguridad al transporte de señalización tanto IPSEC como TLS pueden usarse para proteger los streams SCTP frente a ataques pasivos y activos. Los ataques pasivos se dan cuando una entidad maliciosa solo es capaz de leer paquetes de la red sin poder modificarlos o escribir nuevos paquetes. Este tipo de ataques puede violar el secreto de las comunicaciones, por ello se requiere **confidencialidad**. Los ataques activos se dan cuando una entidad maliciosa es capaz de leer y escribir paquetes en la red comprometiendo la integridad de la comunicación. Por ello, es necesario proporcionar **integridad**. En la RFC 3788 [3] hay una discusión acerca de cuáles son los requisitos para el uso de IPSEC o TLS para proporcionar confidencialidad e integridad al transporte de la señalización. A ambos protocolos se le solicita que provean características de seguridad como autenticación de las partes, integridad, confidencialidad y protección contra los ataques de repetición. Además se requiere no repudio y protección contra usos no autorizados o no apropiados además de protección contra ataques de denegación de servicio. La RFC 3788 analiza también los distintos inconvenientes, desde el punto de vista de la seguridad, de usar TLS y IPSEC. TLS proporciona un nivel de seguridad mayor puesto que ayuda a separar criptográficamente los contextos de diferentes aplicaciones, situadas en el mismo nodo, que transmiten mensajes de señalización. No obstante se evidencian algunos defectos de seguridad en la implementación de TLS que dejan espacio a posibles mejoras. En la sección II-C se analizan los aspectos del uso propuesto de TLS sobre SCTP que lastran el rendimiento. Este articulo presenta algunas modificaciones a la implentación de TLS sobre SCTP definida en la RFC 3436 [4] con el propósito de obviar los problemas de seguridad existentes y mejorar las

prestaciones ofrecidas. Las propuestas de mejora a TLS sobre SCTP serán introducidas en la sección III. La sección actual explica el uso de TLS sobre SCTP como define la RFC 3436 y las consideraciones para el transporte de la señalización explicadas en [3].

II-A. Características del protocolo TLS

TLS negocia el mecanismo de autenticación y efectúa dicha autenticación durante la fase de handshake. Los mensajes de handshake de TLS pueden ser extendidos como se describe en [5] y en [6] para dotar a TLS de funcionalidad nueva como puede ser la negociación de confianza [7]. El mecanismo genérico de extensión se basa en los mensajes de Client y Server Hello y ha sido diseñado para ser compatible con versiones anteriores del protocolo. Este método será usado para aumentar las prestaciones de TLS sobre SCTP. A continuación describimos las características del protocolo TLS.

TLS define el *TLS Record Protocol* como un protocolo estructurado a capas. Los clientes de esta capa proporcionan campos donde se declara longitud, descripción y contenido. Los mensajes pertenecientes a una determinada capa de TLS son identificados por un identificador o content type, lo cual hace posible que se transmitan en la conexión cifrada datos de diferentes capas en paralelo. Los consiguientes mensajes del protocolo TLS Record son fragmentados, opcionalmente comprimidos, cifrados y autenticados utilizando un algoritmo de MAC (Message Authentication Code) como define el contexto negociado o *Connection State* negociado durante la fase de handshake.

Connection State especifica, entre otras cosas los algoritmos de MAC, de cifrado, de compresión así como la clave master-secret. El estándar TLS define cuatro clientes del protocolo *TLS Record Protocol*: el protocolo de handshake que negocia y entrega las claves necesarias al intercambio seguro de mensajes; el protocolo de alerta que produce mensajes comunicando la gravedad de una alerta y su descripción; el protocolo *change cipher spec* que señala cambios en el *Connection State* actual; y el protocolo de aplicación, por ejemplo HTTP, FTP o SMTP. El valor de *Connection State* se negocia durante la fase de handshake. Desde los datos aleatorios enviados por las dos partes se deduce una clave para cifrado simétrico utilizando una función pseudo aleatoria (PRF) y se intercambia dicha clave usando un algoritmo de cifrado asimétrico (RSA o Diffie Hellman). Una vez derivada la clave, el nuevo *Connection State* se usa para proteger el protocolo TLS record proveyendo confidencialidad e integridad.

Handshake: El cliente comienza la fase de handshake enviando un mensaje de *Client Hello* y un conjunto de extensiones opcionales. En este mensaje, el cliente proporciona los datos aleatorios, una marca de tiempo, el identificador de sesión, un conjunto de "cipher suites" y los mecanismos de compresión que soporta. El servidor contesta con un mensaje de *Server Hello*, proporcionando a su vez los datos aleatorios y seleccionando los "cipher suites" y el método de compresión elegido a partir de los propuestos por el cliente. El mensaje de *Server Hello* puede acompañarse de un conjunto de extensiones opcionales. El servidor envía opcionalmente un certificado o un mensaje de *ServerKeyExchange* que será usado para realizar el intercambio seguro de claves (usando el certificado o Diffie Hellman). Opcionalmente el servidor puede solicitar un certificado al cliente (enviando un mensaje de *CertificateRequest*).

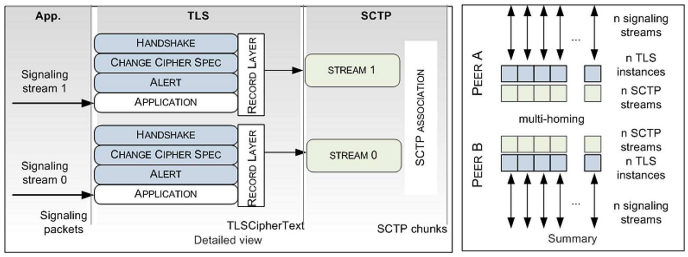


Figura 1. Uso de TLS sobre SCTP según RFC 3436.

El cliente envía su certificado, si le ha sido solicitado, y envía el mensaje obligatorio de *ClientKeyExchange* para intercambiar la clave "pre-master key" del cual se derivará la clave simétrica final que protegerá el canal. Los mensajes de *ChangeCipherSpec* y de *Finished* señalan que hay una nueva clave simétrica lista para ser usada en el *Connection State*, dicha clave se usará para proteger los mensajes de aplicación. El mensaje de *Finished* es enviado sobre el canal cifrado. Cuando este mensaje es recibido, se comprueba su código MAC, que incluye un resumen de los mensajes de la fase de handshake, de modo que implícitamente se puede verificar la integridad del handshake.

II-B. TLS sobre SCTP

SCTP es un protocolo orientado a mensaje con soporte para múltiples flujos, para evitar el "head-of-line" blocking, y multihoming para proporcionar redundancia en la red. Además SCTP permite la configuración dinámica de las direcciones IP. TLS necesita un protocolo orientado a flujo de datos (como TCP) pata transportar los mensajes TLSCipherText. De todos modos TLS puede ser usado sobre SCTP, aunque sea orientado a mensaje, como se describe en la RFC 3436 [4]. En esta especificación se define como usar las capacidades multiflujo y de orientación a mensajes de SCTP para TLS. TLS no es capaz de manejar diferentes valores de MTU si no que se limita a proporcionar a los mensajes TLSCipherText cifrado, protección de integridad (y opcionalmente compresión) con una longitud máxima de 18437 bytes y pasarlos a la subyacente capa de transporte. Quien se encarga de fragmentar y volver a unir los mensajes para evitar la fragmentación a nivel IP es el protocolo de transporte. Para poder usar TLS sobre SCTP, éste tiene que implementar un API que permita la entrega parcial de los mensajes.

En lo que concierne a los flujos SCTP, una asociación SCTP habilitada para el transporte de mensajes TLS entre dos extremos A y B se compone de n flujos bidireccionales (no siendo posible utilizar flujos unidireccionales). La [4] requiere realizar un handshake TLS por cada flujo bidireccional existente entre A y B, siendo posible realizar un handshake completo para el primer flujo y uno abreviado, aprovechando la sesión creada por el primero, para el resto de flujos. Por esta razón se instancia un contexto TLS diferente por cada flujo SCTP. (véase Fig. 1).

II-C. Defectos de TLS sobre SCTP

En esta sección vamos a explicar cuales son los defectos de la definición de TLS sobre SCTP tal como se especifica en la RFC 3436 [4]. Según lo que ha explicado hasta el momento, si la primera sesión de TLS se resume durante el resto de los handshakes, resulta en un nuevo *Connection State* por stream pero con la misma clave "master key" (usada n veces una por cada flujo). La duración de la fase de establecimiento de la sesión TLS sobre la asociación SCTP depende pues de la complejidad de los handshakes, sean esos completos o abreviados, de las prestaciones de la red, los retrasos de "round-trip time", ancho de banda...

Con un elevado número de flujos la cantidad de datos total por procesar podría crecer de forma no lineal debido a que la RFC 3436 impone una instancia TLS por flujo, lo que requiere un cifrador por flujo. Sin embargo, pese a que el hecho de tener una clave distinta por cada flujo permite aislar criptográficamente cada flujo de los demás tener varias instancias de TLS no añade un nivel de seguridad mayor puesto que es la misma aplicación la que maneja todos esos flujos.

Respecto a la seguridad, la RFC 3388 [3] reúne los requisitos para TLS sobre SCTP en el ámbito de transporte de señalización. Esta recomendación evidencia cómo TLS sobre SCTP, en la configuración propuesta en la RFC 3436, deja desprotegidos los chunks de control. SCTP utiliza los chuncks de control para comunicar la organización de la carga de los paquetes o para identificar a qué flujo pertenece cada paquete. TLS sólo protege la carga de los paquetes de forma que las cabeceras de los protocolos inferiores son enviadas en claro. Este problema es común para los protocolos que proveen seguridad a nivel de trasporte pero, en el caso de SCTP, el problema adquiere una mayor relevancia dado que, además que las propias cabeceras de SCTP, también los chuncks de control, usados para controlar los flujos internos, se envían sin cifrar. Esta situación hace posible un nuevo ataque de DOS que puede realizarse a nivel de flujo.

III. Enfoque propuesto

Esta sección explica nuestra propuesta de modificación de TLS sobre SCTP para solucionar los problemas anteriormente comentados. Para entender las motivaciones de nuestro enfoque vamos a explicar mejor las similitudes entre SCTP y TLS. En SCTP, los chuncks pertenecientes a varios flujos son transportados juntos en un mismo paquete, por esta razón los flujos no son más que una herramienta para distinguir tipos de datos diferentes a nivel de aplicación, es decir, no implica que sean transportados de forma independiente. Por otro lado, TLS usa la capa de registro (record) para fragmentar los bloques de información en registros TLSPlaintext antes de cifrarlos partiendo los datos en trozos de 2^{14} bytes o menos. Los límites de los mensajes del cliente no son preservados en esta capa, al igual que SCTP no preserva la organización en streams al componer los mensajes que agrupan los chunks. En TLS varios mensajes del mismo tipo (marcados con el mismo "ContentType") pueden ser transportados en un único registro TLSPlaintext o un único mensajes puede ser fragmentado en varios registros.

Resulta sencillo identificar un solapamiento entre TLS y SCTP en lo que a streams se refiere. TLS distingue entre clientes de la capa de registro usando un identificador llamado "content type" y los limites de sus mensajes desaparecen durante la fragmentación de los paquetes. SCTP utiliza flujos para distinguir entre la información de distintas aplicaciones y también mezcla chunks que llegan de diferentes flujos en un solo paquete. Además, puesto que TLS elimina los límites entre los mensajes de los clientes, cuando SCTP se utiliza con TLS, su orientación al mensaje se mantiene solo hasta el nivel de los paquetes TLSCipherText. En este caso los flujos de SCTP proporcionan la misma funcionalidad de la capa de registro de TLS. Los mensajes TLSCipherText son definidos a continuación:

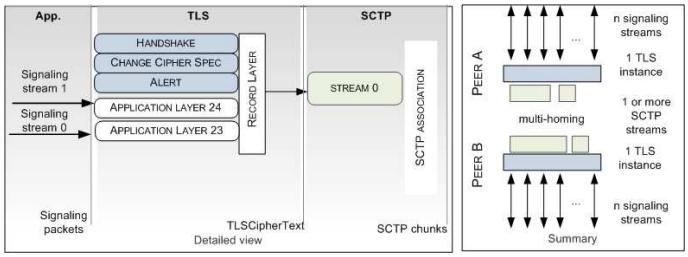


Figura 2. Detalle de uso de TLS modificado sobre SCTP.

```
enum {
    change_cipher_spec(20), alert(21), handshake(22),
    application_data(23), (255)
} ContentType;

struct {
    ContentType type;
    ProtocolVersion version;
    uint16 length;
    opaque fragment[TLSPlaintext.length];
} TLSPlaintext;
```

Actualmente TLS utiliza la capa de registro para multiplexar información procedente de los protocolos de aplicación, handshake, change cipher spec y alert pero es posible añadir más capas. Las recomendaciones definen el "content type" como un tipo numerado variable en un rango desde 0 hasta 255 los valores desde 20 hasta 23 están ya asignados a los clientes estándar. Por esta razón, hemos modificado TLS para permitir la negociación de capas de aplicación adicionales. Una vez negociadas, estas nuevas capas pueden proporcionar la misma funcionalidad de los flujos SCTP (véanse Fig. 2). Nuestra implementación cumple con los siguientes objetivos:

- Separación de la señalización en flujos. Cada flujo es enviado sobre una capa diferente
- Protección de la información de carga y de control. TLS cifra también las cabeceras TLSPlainText así que solo las cabeceras SCTP quedan desprotegidas. Eliminamos así la posibilidad de un ataque DOS sobre un solo flujo.
- Se evitan las interferencias con el soporte multi-homing. Nuestras modificaciones no obstaculizan el multi-homing
- Mejora de las prestaciones por usar una sola instancia de TLS

Una aplicación puede usar TLS sobre SCTP utilizando una única instancia de TLS. SCTP puede usar uno o más flujos, la aplicación puede usar hasta $255-23 = 232$ capas de TLS para organizar su trafico. En el próximo párrafo se explicará como TLS ha sido extendido para soportar estas funcionalidades.

III-A. Extensión TLS: extensiones para capas adicionales

La extensión para el protocolo de handshake de TLS aquí descrita permite a los extremos de una conexión negociar negociar el número de capas a usar durante la sesión TLS. Esta negociación se realiza durante la fase de handshake por acuerdo mutuo. Como se describió en [5] y [8], el handshake TLS incluye extensiones resistentes a ataques Man-in-the-middle pero no a ataques DOS. De todos modos, una vez finalizada la fase de handshake los ataques de DOS ya no son posibles.

El intercambio de mensajes propuesto para la negociación de las capas adicionales se realiza a continuación junto con los mensajes intercambiados habitualmente en TLS (* marca los mensajes opcionales):

```
Client                                        Server
--
ClientHello
<LayerExtension*>          -------->
                                        ServerHello
                                        <LayerExtension*>
                                        Certificate*
                                        ServerKeyExchange*
                                        CertificateRequest*
                           <--------    ServerHelloDone
Certificate*
ClientKeyExchange
CertificateVerify*
[ChangeCipherSpec]
Finished                    -------->
                                        [ChangeCipherSpec]
                           <--------    Finished
Handshake ends.
Application data can be sent using multiple layers

Application Data [layer 23]<--->Application Data [layer 23]
Application Data [layer 24]<--->Application Data [layer 24]
       ...                                        ....
Application Data [layer  n]<--->Application Data [layer  n]
```

Vamos a describir las modificaciones al handshake original de TLS:

1. El cliente señala el uso de capas adicionales con el envío del campo "LayerExtension" al final del mensaje de *Client Hello*. Este mensaje lleva el número de capas que el cliente desea utilizar (n_1).
2. Si la otra parte entiende la extensión, entonces envía al cliente la misma extensión conteniendo el numero de capas que serán utilizadas (n_2 con $n_2 \leq n_1$) al final del mensaje de *Server Hello*.
3. El handshake TLS continúa como se define en el estándar hasta establecer un canal seguro, después el Server envía el mensaje de *Finished*
4. Llegados a este punto, la aplicación puede simultáneamente enviar y recibir tráfico sobre capas distintas compartiendo la misma instancia TLS

Se han definido la extensión LayerExtension como se detalla a continuación:

```
struct {
    LayerRanges LayerRange<0..2^16>;
}LayerExtension

struct {
    opaque     identifier<0..2^16-1>;
    uint8     first;
    uint8     count;
} LayerRange
```

El campo *LayerExtension* lleva una lista de rangos de capas (*LayerRange*). Un rango se define por el número de capas solicitado y por el número de la primera capa del rango que se solicita asignar. Es posible asignar grupos de capas disjuntos, un solo grupo o una sola capa. El identificador puede ser un número, una URI o una descripción textual.

La metodología de negociación descrita utiliza el extensiones tal y como se define en el protocolo de handshake de TLS descrito en la RFC 4346 donde se describe la compatibilidad hacía atrás de las extensiones. Un servidor que no implemente el tipo de extensión requerida para la negociación de capas adicionales ignorará la extensión recibida de modo que el cliente, al recibir el mensaje *Server Hello* sin la extensión esperada, pasará al proceso de handshake estándar. En este caso la conexión TLS continuaría de forma estándar dando lugar a una conexión donde se utiliza una única capa para el transporte de datos de aplicación.

IV. Implementación y resultados

Con el objetivo de demostrar que nuestra propuesta aporta una mejora real a la implementación de TLS sobre SCTP propuesta en la RFC 3436, se ha desarrollado una infraestructura de pruebas con ambas implementaciones utilizando la librería OpenSSL [9] como punto de partida. Dicha infraestructura permite a dos extremos efectuar una negociación e intercambiar datos usando TLS sobre SCTP con o sin capas adicionales y recoge medidas de los tiempos de transmisión y de la carga. Han sido realizados dos experimentos:

- Para el primer experimento se creó una asociación SCTP con N flujos entre los extremos. Después se realizó un handshake completo por cada flujo SCTP, como se propone en la RFC 3436. Una vez finalizado el último handshake, ambos extremos eran capaces de comunicar de forma segura e independiente sobre cada flujo bidireccional. Por cada uno de esos flujos, se enviaron 4096 KBytes de datos de señalización simultáneamente. La Fig. 1 puede ayudar a entender este escenario.
- En el segundo experimento se creó una asociación SCTP con un solo flujo entre las aplicaciones cliente y servidor. Después se realizó un handshake TLS extendido usando la extensión para capas adicionales (LayerExtension) solicitando N capas al servidor. El handshake modificado ha sido descrito en la sección III-A. Una vez finalizado el handshake, quedanban disponibles N capas TLS para la transmisión independiente de datos sobre el mismo flujo SCTP. Simultáneamente se envió sobre cada una de dichas capas 4096 Kbytes de datos de señalización. Véase la Fig. 2.

El objetivo de los dos experimentos es medir el tiempo necesario para realizar la trasferencia completa de una carga repartida por igual entre los flujos de comunicación establecidos por los extremos. Todas las medidas se realizan a partir del establecimiento de la asociación SCTP e incluyen el tiempo necesario para la realización de la fase de handshake. Una trasferencia se considera completada en el instante en el cual el ultimo paquete de datos transmitido es recibido por el extremo final de la asociación. La trasmisión de datos se efectúa de forma paralela tanto en el caso 1 como en el caso 2. Ambos experimentos han sido repetidos varias veces aumentando en 5 el número de flujos de señalización. En el primer experimento no ha sido posible superar el umbral de

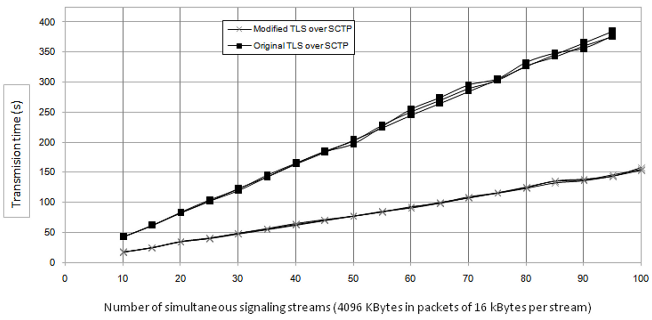


Figura 3. Resultado de las pruebas. Comparación con la RFC 3436.

los 95 flujos simultáneos. En el segundo experimento se han detenido las pruebas al llegar a 100 flujos puesto que no se tendrían datos de comparación más allá de este umbral. Se ha medido tanto el tiempo de transmisión como el overhead de envío. En lo que respecta a la carga se ha medido la cantidad de bytes totales enviados entre las dos aplicaciones. De esta forma se tienen en cuenta los *chunks* de control, las cabeceras de los *chunks* de datos componentes los flujos, las cabeceras generales SCTP y las cabeceras enviadas por el protocolo TLS. Para ambos experimentos se han obtenido datos de carga similares aunque el segundo experimento introduzca un reducción respecto al primero cerca del $0,15\,\%$. En cambio, en términos de tiempo total de transmisión, el segundo experimento introduce una mejora sustancial en las prestaciones respecto al primero como se puede ver en la Fig. 3.

En la Fig.3 se puede ver como el tiempo de transmisión global aumenta linealmente con el número de los flujos de señalización utilizados para ambos experimentos. Los resultados de nuestra solución presentan un offset inicial más reducido y una pendiente $2,5$ veces más pequeña que TLS sobre SCTP de la RFC 3436. Como nos esperábamos, la presencia de una instancia TLS por cada flujo introduce un retardo considerable.

V. Trabajos relacionados

Existen varios trabajos de relevancia que extienden TLS a través del uso intensivo de extensiones como se propone en [5], [8] y [6]. Hay Extensiones TLS para el soporte de nuevas funcionalidades como autorización [10], [11], autenticación/autorización avanzadas [12] y también negociación de confianza [7]. La mayoría de las extensiones proporcionan nuevas funcionalidades al handshake de TLS permitiendo la negociación de nuevos parámetros, además realizan operaciones basadas en dichos parámetros que tienen que ser manejadas por la capa de handshake.

La utilización de nuevas capas para realizar operaciones de forma independiente de las capas estándar fue introducida en [7]. Este artículo propone la negociación de nuevas funcionalidades que han de ser manejadas por diferentes capas del protocolo sin introducir sobrecarga en capas estándar, como la de handshake, que no han sido diseñadas para estos propósitos. De esta manera, se evitan errores y se reduce la complejidad. Nuestra propuesta sigue el mismo principio usando capas para transmitir información de forma independiente de otras capas, emulando así los flujos SCTP. Se trata de un enfoque original y nunca antes propuesto al problema de transmitir flujos independientes de información sobre un canal seguro.

VI. Conclusiones

El presente artículo discute los problemas de TLS sobre SCTP tal y como se proponen en la RFC 3436 con un enfoque especial en las prestaciones. La RFC 3436 describe la utilización de una instancia TLS por cada flujo SCTP resultando en un enorme tiempo de establecimiento de conexión y un mal empleo de los recursos. A pesar de que esta recomendación propone realizar un handshake completo para el primer flujo y uno abreviado para lo siguientes (resumiendo la primera sesión TLS) una instancia de TLS por flujo es innecesaria. Esta sobrecarga extra es un coste inútil puesto que no introduce ninguna mejora en la seguridad.

En este artículo se propone el uso de capas del protocolo TLS, del mismo modo en el que se usan los flujos SCTP, para reutilizar la misma instancia TLS de forma que se pueda llevar más de un flujo de señalización y mejorar de modo considerable las prestaciones. Nuestra propuesta usa una sola instancia TLS pues un solo handshake es suficiente para llevar múltiples flujos de señalización sobre un canal seguro usando un solo contexto y un solo conjunto de claves. Además, se ha implementado y testado frente a la implementación propuesta en la RFC 3436 y, como se ha puesto de manifiesto en la sección IV, la mejora es considerable (manteniéndose además las propiedades deseadas de SCTP como evitar el head-of.-line blocking).

En estos momentos, está mejora se está portando a OpenIMS Core para mejorar las prestaciones del interfaz Za que comunica con otros proveedores a través de un interfaz seguro.

Agradecimientos

Este trabajo ha sido financiado parcialmente por la ayuda a Grupos Investigación CAM CCG08-UC3M/TIC-4479.

Referencias

[1] Stewart, R.: Stream Control Transmission Protocol. RFC 4960 (Proposed Standard) (2007)
[2] Ong, L., Yoakum, J.: An Introduction to the Stream Control Transmission Protocol (SCTP). RFC 3286 (Proposed Standard) (2002)
[3] Loughney, J., Tuexen, M., Pastor-Balbas, J.: Security Considerations for Signaling Transport (SIGTRAN) Protocols. RFC 3788 (Proposed Standard) (2004)
[4] Jungmaier, A., Rescorla, E., Tuexen, M.: Transport Layer Security over Stream Control Transmission Protocol. RFC 3436 (Proposed Standard) (2002)
[5] Dierks, T.: The tls protocol. Technical Report RFC 2246, IETF TLS Working Group (1999)
[6] Blake-Wilson, S.: Transport layer security (tls) extensions. Technical Report RFC 3546, IETF TLS Working Group (2003)
[7] Sánchez, D.D., Marín, A., Mendoza, F.A., Campo, C., Cortes, A., García-Rubio, C.: Trust negotiation protocol support for secure mobile network service deployment. In Mammeri, Z., ed.: MWCN/PWC. Volume 284 of IFIP., Springer (2008) 271–282
[8] Dierks, T., Rescorla, E.: The transport layer security (tls) protocol. version 1.1. Technical Report RFC 4346, IETF TLS Working Group (2006)
[9] Henson, S.: Openssl project (2009) http://www.openssl.org.
[10] Farrell, S.: Tls extensions for attributecertificate based authorization. Technical Report draft-ietf-tls-attr-cert-01.txt, IETF Transport Layer Security Working Group (1998)
[11] Brown, M., Housley, R.: Transport layer security (tls) authorization extensions. Technical Report draft-housley-tls-authz-extns-07.txt, IETF (2006)
[12] Hess, A., Jacobson, J., Mills, H., Wamsley, R., Seamons, K., Smith, B.: Advanced client/server authentication in tls (2002)

Un nuevo ataque a TCP para redes de radios cognitivas

Olga León*, Juan Hernández-Serrano* y Miguel Soriano*†
*Universitat Politècnica de Catalunya (UPC), España
†Centre Tecnològic de Telecomunicacions de Catalunya (CTTC)
{olga,jserrano,soriano}@entel.upc.edu

***Resumen*—Los dispostivos de radios cognitivas emergen como una prometedora tecnología que ha de permitir un mejor uso del espectro electro-magnético. Estos dispositivos se caracterizan por ser capaces de observar y entender su entorno, y cambiar consecuentemente su modo de operación. Sin embargo, estas propiedades "cognitivas" conllevan nuevos retos de seguridad. En este artículo se presenta un nuevo ataque a las conexiones TCP en redes de radios cognitivas, se proponen soluciones para mitigarlo y se evalúa el impacto del mismo con y sin contramedidas.**

***Palabras Clave*—red de radios cognitivas, CRN, radio cognitiva, CR, TCP, ataque, seguridad**

I. Introducción

Con el auge de las aplicaciones inalámbricas, el espectro radio-eléctrico se está convirtiendo en un recurso escaso y muy apreciado. Las agencias reguladoras de cada estado dictaminan en que frecuencias pueden operar determinados sistemas, es decir, que asignan o licencian de forma estática partes del espectro para servicios específicos. Esta asignación de frecuencias es de naturaleza estática y conlleva, por tanto, a una utilización muy poco eficiente del espectro limitando en gran medida las posibilidades de uso del mismo. De hecho, las bandas frecuenciales por debajo de 3GHz están ampliamente infrautilizadas en, al menos, la Comunidad europea, Estados Unidos de América, China y Japón; con usos que para un determinado lugar y momento rara vez sobrepasan el 20 % de los recursos disponibles.

Un dispositivo inalámbrico convencional puede tan sólo acceder a un área del espectro radio, pero un dispositivo de radio cognitiva (*cognitive radio* - CR) puede también detectar e identificar "espacios blancos" -o áreas vacantes- en el espectro que puedan utilizarse para establecer una comunicación. Los dispositivos CRs son radios inteligentes que tienen la habilidad de sentir su entorno, aprender de su historia y tomar decisiones inteligentes para ajustar sus parámetros de acuerdo a las condiciones observadas. Las redes de radios cognitivas (*Cognitive radio networks* CRNs) emergen como una posible solución al problema de infrautilización del espectro, pues de forma inteligente encuentran bandas de operación disponibles, y lo que es más importante, sin interferir en el funcionamiento normal de los sistemas licenciados. Este hecho permite usar el espectro a dispositivos no licenciados, o usuarios *secundarios*, sin afectar el funcionamiento típico de los usuarios licenciados o *primarios*.

Las capas física y de enlace de las CRNs son muy diferentes de las redes inalámbricas convencionales. Características propias de las CRNs, como la observación del entorno de forma cooperativa y los mecanismos de coexistencia con otras CRNs o con sistemas primarios, producen nuevas e interesantes implicaciones de seguridad. A pesar de todo, este tópico ha recibido, hasta el momento, mucha menos atención que otras áreas de investigación relacionadas con las CRNs. De todas formas, las escasas contribuciones en el estado del arte se centran principalmente en dos amenazas propias de las CRNs: 1) ataques basados en la emulación de usuarios primarios, y 2) ataques a la función de objetivos del protocolo de aprendizaje de los dispositivos CR.

El ataque de emulación de usuario primario (*primary user emulation* - PUE), citado por primera vez en [1], se basa en el hecho de que a las CRs o usuarios secundarios se le permite tan sólo operar en bandas licenciadas si no producen ninguna clase de interferencia a los usuarios primarios. Así pues, las CRs deben observar continuamente el medio para detectar la posible presencia de usuarios primarios y, en tal caso, conmutar a otra banda. Este hecho introduce un agujero de seguridad en el sistema: un atacante puede evitar que la CRNs utilicen determinadas bandas transmitiendo una señal similar a una primaria.

Los ataques a la función de objetivos (*Objective function attacks* - OFA) [2] pretenden alterar el funcionamiento del protocolo de aprendizaje de las CRs. En una CRN, cada dispositivo controla una serie de parámetros para mejorar el funcionamiento de la red. La elección de parámetros se hace, a menudo, mediante un algoritmo de inteligencia artificial que introduce pequeñas alteraciones de configuración hasta obtener los valores óptimos que maximizan una función de objetivos preestablecida. Un atacante con concocimiento total o parcial del protocolo de aprendizaje podría alterar las condiciones para obtener un beneficio personal. Por ejemplo, un caso naïve sería aquel en que el atacante genera interferencias siempre y cuando se establezca cifrado en la comunicación. Probablemente, el algoritmo de inteligencia de las víctimas decidirá que un parámetro óptimo es no cifrar y permitirá la posterior interceptación de datos por parte del atacante.

En este artículo se identifica una nueva amenaza de seguridad propia de las CRNs. Se trata de un ataque entre-capas (*cross-layer*) desde la capa física/enlace que afecta a las comunicaciones TCP de la capa de transporte. El ataque se basa en

forzar cambios de frecuencia (*handoffs*) de la CRN afectando especialmente a las conexiones TCP actuales. La sección II describe de forma detallada el ataque. A continuación, en la sección III se proponen soluciones para mitigar al mismo. Después, se presenta una evaluación del impacto del ataque con y sin contramedidas en la sección IV. Finalmente, en la sección V se presentan las conclusiones de este trabajo.

II. Ataque Lion: ataque contra el TCP en CRNs

Se define el ataque "Lion" como un ataque *jamming* orientado a reducir el caudal TCP mediante la provocación intencionada de *handoffs*. Puede ser visto, por lo tanto, como un ataque *cross-layer* desde la capa física/enlace a las comunicaciones TCP de la capa de transporte.

En la literatura, se han descrito otros ataques *cross-layer* al TCP. Uno de los más conocidos es el ataque "Jellyfish" [3], cuyo objetivo es reducir el caudal TCP en los enlaces atacados alterando parámetros de la comunicación, como retardos y pérdidas. Produce, como consecuencia, un reducción del TCP puesto que indirectamente altera algunas de sus funciones, como las medidas de *round trip time* (RTT), estimaciones del *retransmission timeout* (RTO), *slow start* o *congestion avoidance*. La forma en que lo consigue es simplemente desordenando, eliminando a alterando el retardo de los paquetes que circulan por un determinado enlace. Las principales diferencias entre el ataque Lion y el Jellyfish son, que en el primero: 1) la fuente del ataque es externa y no necesita formar parte activa de la red; y 2) la degradación del caudal TCP radica en los *handoffs* de canal propios de las CRNs.

A los usuarios de una CRN, como secundarios de uso del espectro, sólo se les permite usar las bandas licenciadas si no interfieren las comunicaciones de los usuarios primarios. Si se detecta la presencia de un primario, los secundarios deben cambiarse a otra banda disponible o, lo que es lo mismo, ejecutar un *handoff*. Este proceso genera un retardo considerable hasta que se recupera la comunicación (de un orden máximo de segundos) pues comprende la observación del medio para detectar canales vacantes (oportunidades), escoger la mejor oportunidad y conmutar a su frecuencia.

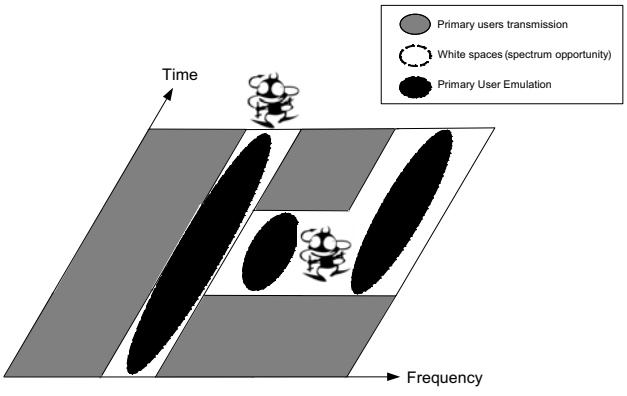


Figura 1. Ataque PUE

Un usuario malicioso que intenta degradar una conexión TCP de un usuario de la CRN puede trasmitir intencionadamente una señal en la misma banda emulando la de un usuario primario (ver figura 1), y forzar por tanto a que la CRN ejecute un *handoff* y cambie a otro canal. Como la capa de transporte no es consciente de la interrupción física de la conexión, continúa enviando segmentos que son almacenados en una cola de transmisión en capas inferiores. Así, los segmentos TCP pueden ser retardados o incluso perdidos durante el *handoff* produciendo una reducción del caudal TCP. Cuanto más larga es la duración del *handoff*, más drástica la reducción de caudal.

TCP mantiene un temporizador de retransmisión (RTO) para cada segmento TCP cuyo valor proviene de una estimación basada en las medidas del RTT tomadas durante la conexión. Si el RTO de un determinado segmento expira sin haberse recibido una confirmación (*acknowledgement* - ACK), TCP considera que el segmento se ha perdido por congestión en la red, dobla su valor de RTO, lo retransmite y reduce su ventana de congestión a un segmento, reduciendo por tanto el caudal TCP. Que expire el RTO se puede deber a congestión, pero también a un incremento repentino del RTT. De esta forma, un periodo de *handoff* suficientemente grande, como es el caso de las CRNs, conllevará con una alta probabilidad la expiración de los temporizadores y la consecuente degradación de caudal TCP.

De hecho, un fallo en la retransmisión de un segmento debido a la persistencia del *handoff* puede llevar a una situación incluso peor. Con cada intento infructuoso el tiempo entre retransmisiones se dobla siguiendo una progresión exponencial, lo que, si hay muchos fallos, puede llevar al emisor a permanecer inactivo por periodo muy largo de tiempo. Desde nuestro punto de vista, este hecho sugiere la necesidad de utilizar mecanismos *cross-layer* para que los protocolos de transporte sean conscientes de las condiciones de red.

La figura 2 muestra dos posibles escenarios donde se produce un *handoff* mientras se está transmitiendo por una conexión TCP. Todos los segmentos dentro de la ventana de transmisión TCP se consideran como perdidos debido a la expiración del temporizador de retransmisión. Además, la ventana de congestión se reduce a un sólo segmento que poco a poco irá aumentando a medida que no se midan problemas de congestión (nuevos temporizadores de retransmisión expirados). En la figura 2a, el *handoff* termina antes de que se produzca la retransmisión y, por lo tanto, el emisor puede recuperarse de las perdidas en un periodo relativamente corto de tiempo. La figura 2b muestra, en cambio, el peor caso posible en cual las consecuentes retransmisiones fallan también. Puesto que el temporizador de retransmisión se dobla cada vez que expira, el fallo de retransmisiones consecutivas tiende a dejar en espera al emisor por un periodo muy largo de tiempo.

El ataque Lion puede convertirse en un ataque de denegación de servicio (*denial of service* - DoS) si el atacante puede predecir o conocer los nuevos parámetros de transmisión que se utilizarán en la CRN después del *handoff*. Por ejemplo, supongamos que un cada vez que una CRN conmuta a un nuevo canal, el atacante, a sabiendas de qué canal será, fuerza otro *handoff*, p.e. mediante transmisión PUE. Evidentemente

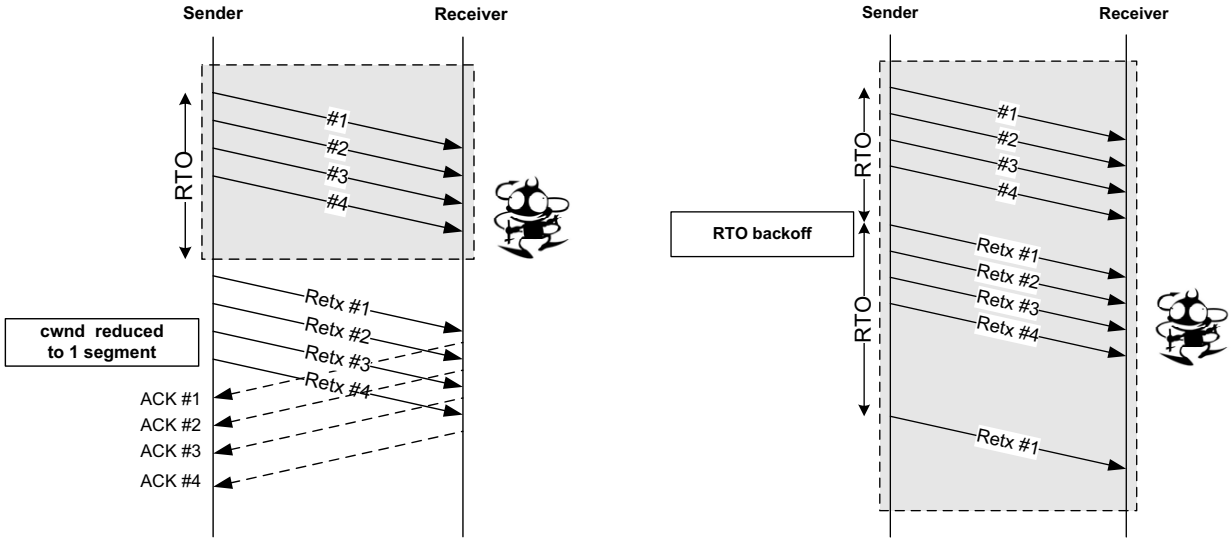


Figura 2. Efecto del handoff sobre una conexión TCP

en este caso la CRN entrará en un bucle de *handoffs* que inhabilita por completo sus comunicaciones.

III. Mitigando el ataque Lion

El ataque Lion fuerza a los usuarios de la CRN a cambiar la frecuencia de transmisión (*handoff)*, introduciendo un retardo considerable hasta que las transmisiones se reanudan. Para evitar que las capas superiores de la pila de protocolos malinterpreten la información y reaccionen inadecuadamente, deben ser conscientes de lo que ocurre a niveles más bajos. Del mismo modo, este razonamiento puede aplicarse también a los mecanismos de seguridad, que no deberían centrarse en una única capa de manera aislada. La comunicación entre capas no adyacentes es útil no sólo para mejorar el funcionamiento global de la red sino también para mitigar ataques como el ataque Lion. Para mitigar un ataque en el que se intenta interrumpir una conexión TCP mediante PUE, deben llevarse a cabo dos tareas fundamentales: 1) detectar el ataque e identificar/localizar al atacante; y 2) proporcionar la información necesaria a la capa de transporte para minimizar el impacto del ataque en la operación normal del protocolo.

En la literatura han aparecido varias propuestas *cross-layer* cuyo objetivo es mejorar el rendimiento de TCP en redes inalámbricas, en especial redes ad hoc. Estas propuestas intentan solucionar problemas típicos de entornos inalámbricos como pérdidas de paquetes (no debidas a congestión), cambios repentinos de rutas o pérdidas temporales de conectividad; que afectan negativamente al funcionamiento de TCP debido a su interacción con los mecanismos de control de congestión de TCP. Desde nuestro punto de vista, estas técnicas pueden usarse como pauta para el diseño de nuevos protocolos para CRNs, mejorando el rendimiento y la robustez frente a ataques *cross-layer*. Entre las propuestas existentes cabe destacar Freeze-TCP, una variante de TCP diseñada para mejorar el rendimiento en entornos móviles, donde se producen desconexiones con frecuencia debido a la pérdida de señal o al movimiento de los nodos. En Freeze-TCP, el receptor monitoriza la potencia de la señal recibida para poder predecir desconexiones, en cuyo caso anuncia una ventana de tamaño 0 bytes al emisor antes de que se produzca la desconexión. La recepción de dicha ventana impide al emisor transmitir datos y lo fuerza a entrar en modo ZWP (*Zero Window Probe*), congelando todos los parámetros de transmisión (ventana de congestión, temporizadores de retransmisión, etc.). Cuando la conexión vuelve a estar disponible, el receptor anuncia una ventana de tamaño superior a 0 bytes que permite al emisor continuar con la transmisión. Mediante este mecanismo, es posible evitar pérdidas y retransmisiones innecesarias y por tanto se mejora el rendimiento de TCP.

Para mitigar el efecto de los *handoffs* en CRNs podría utilizarse Freeze-TCP asumiendo que el nodo receptor puede predecir cuando la CRN va a conmutar de canal, i.e., *handoff* proactivo. Sin embargo, Freeze-TCP no considera la variación de las características de la conexión después del *handoff*. Un cambio en los parámetros de transmisión, como frecuencia o codificación, puede implicar a su vez un cambio en el ancho de banda disponible, de manera que puede no ser adecuado mantener los mismos valores en la conexión TCP después del *handoff*. A modo de ejemplo, una reducción del ancho de banda podría provocar múltiples pérdidas si el emisor no reduce su ventana de congestión. Por este motivo, la actual versión de Freeze-TCP no es óptima para este entorno.

Por otro lado, considerando que todos los nodos pertenecientes a una determinada CRN comparten la misma información, el emisor sabe cuando se va a producir el *handoff* y puede congelar sus parámetros sin necesidad de aviso del receptor. Esto llevaría a una nueva versión de TCP que congelase la transmisión de segmentos con dos diferencias fundamentales respecto a Freeze-TCP: 1) el emisor congela la transmisión TCP sin necesidad de esperar a que el receptor

envíe un aviso; y 2) como cada participante tiene información sobre cuál será la siguiente banda de frecuencias en términos de ancho de banda disponible, relación señal a ruido, etc, se modifican los parámetros TCP consecuentemente.

Debe tenerse en cuenta que congelar los parámetros TCP no protege del ataque completamente sino que únicamente lo mitiga. Si el atacante persiste forzando *handoffs* puede llegar a provocar un DoS permanente. Para evitar esta situación , se debe impedir que el atacante detecte rápidamente la siguiente banda de frecuencias a utilizar por la CRN. Asumiendo que el atacante es un dispositivo CR, puede predecir dichas frecuencias de dos modos: 1) mediante detección local; y 2) obteniéndolas a partir de la información de control de la CRN.

En nuestra opinión, la detección por parte de un atacante del siguiente salto de frecuencias despues de un *handoff* es una amenaza limitada, puesto que la elección del mismo se realiza de forma cooperativa entre todos los participantes de la CRN y al atacante sólo observa en un entorno reducido (detección local). Las oportunidades de espectro pueden variar de un lado a otro de la CRN y por ello, la mejor oportunidad obtenida mediante detección local puede diferir de la global. De todos modos, la detección local proporciona al atacante información valiosa, ya que al menos puede determinar qué frecuencias son completamente improbables. Por lo tanto, la mejor defensa frente a estos ataques es disponer de un amplio abanico de bandas no licenciadas con características similares, caso en que la correlación entre la detección local y la decisión global será menor.

A pesar de que las predicciones mediante detección local están limitadas, interceptar la información de control de la CRN puede permitir al atacante conocer el canal al que va a conmutar la CRN. Como consecuencia, es evidente que se debe dotar de seguridad a dicha información. Dado que la información de control es compartida por todos los miembros del grupo, el modo más sencillo y eficiente de hacerlo es mediante el uso de un secreto compartido o clave de grupo. Esta clave permite que cada miembro de la CRN: 1) envíe datos cifrados; 2) descifre los datos recibidos, y 3) se autentique como un miembro de la CRN, dado que el conocimiento de la clave de grupo garantiza que pertenece a dicha CRN. Únicamente los miembros del grupo deben conocer la clave y por lo tanto, dicha clave debe actualizarse cada vez que hay cambios en el grupo. La gestión de claves de grupo (GKM) estudia cómo generar y actualizar el material de claves utilizado para la seguridad de grupo a lo largo de la vida del mismo, centrándose en el dinamismo de grupo [4]. La integración de protocolos eficientes GKM en redes CRNs es, desde nuestro punto de vista, un paso necesario para mitigar el efecto del ataque Lion específicamente y, de forma global para asegurar las CRNs.

Todas las contramedidas presentadas anteriormente pueden mitigar parcialmente los efectos del ataque Lion pero no evitarlos en su totalidad, ya que no pueden tratar con efectividad los ataques DoS o la degradación del canal debida a *jamming*. Por este motivo, es necesario usar en paralelo un sistema de detección de intrusos (*intrusion detection system* - IDS).

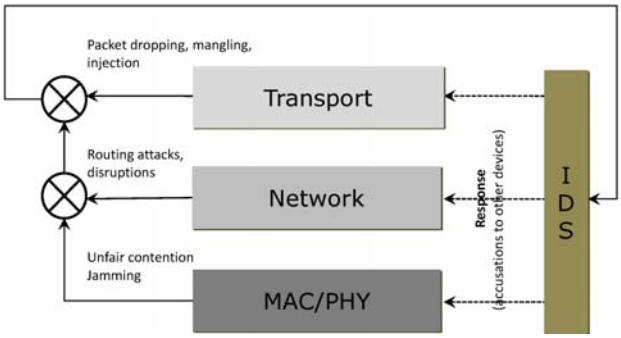


Figura 3. IDS cross-layer

Los IDSs se encargan de monitorizar dispositivos en busca de desviaciones respecto al funcionamiento habitual de un protocolo, permitiendo p.e. detectar la presencia de nodos sospechosos o maliciosos. Sin embargo, la mayoría de las soluciones existentes operan en cada una de las capas de la pila de protocolos de forma independiente. En nuestra opinión, los IDSs para CRNs deben cumplir con los siguientes requisitos para poder mejorar los mecanismos de detección: 1) la monitorización del tráfico debe hacerse de forma cooperativa y distribuida, especialmente para garantizar la correcta localización del atacante; y 2) debe garantizarse la interacción entre capas para mejorar la detección de ataques *cross-layer*, p.e. el ataque Lion puede detectarse combinando información de la capa física y de transporte. A pesar de que existen diversas propuestas [5]–[7] que satisfacen los requisitos mencionados su aplicación a CRNs constituye todavía un reto.

El estado del arte presenta IDSs basados principalmente en dos técnicas: mal-uso y detección de anomalías. En el primer tipo, se construye un patrón o modelo específico, denominado "firma" para cada ataque conocido. Los IDS pertenecientes al segundo tipo, se basan en considerar sospechosa cualquier desviación respecto a la operación normal del protocolo.

Las CRNs abarcan una gran variedad de tecnologías y protocolos y por ello, están potencialmente expuestas a múltiples y diversos ataques. Como consecuencia, los IDSs basados en técnicas de detección de mal-uso pueden resultar poco prácticos debido a la dificultad existente en determinar, distribuir y actualizar las firmas de todos los posibles ataques. Por otro lado, las técnicas basadas en detección de anomalías podrían aplicarse pero son propensas a tasas altas de falsos positivos, especialmente en entornos de naturaleza poco controlada como es el caso de las CRNs. Por ello, queda patente la necesidad de nuevos esquemas que ayuden a diagnosticar con precisión ataques maliciosos en CRNs. Concretamente, es necesario definir un conjunto de características que permitan correlar la información de distintas capas para poder modelar el comportamiento normal de los nodos (mirar figura 3).

En CRNs, los nodos sólo pueden monitorizar el tráfico de red dentro de su rango de transmisión. Dado que normalmente, las observaciones varían dependiendo de su localización, una colisión local podría impedir que un nodo observe el medio durante un cierto periodo de tiempo. Este hecho lleva a ini-

cialmente confiar en un nodo que afirma no poder monitorizar, pero abre la puerta a comportamientos egoístas o mentirosos. Además, un nodo malicioso podría falsificar información intencionadamente con el fin de interrumpir o alterar el funcionamiento de la red. Por este motivo, se hace necesario un sistema de reputación para desarrollar un modelo de confianza [8] en CRNs. Este modelo debería beneficiarse de la redundancia de la red, puesto que el *feedback* de los distintos participantes puede ayudar a detectar con mayor facilidad la fuente del ataque. Asimismo, debería implementarse un análisis global de los mecanismos de monitorización/observación para poder determinar si los nodos están cooperando o no, y establecer niveles de confianza para los datos recolectados.

IV. Evaluación

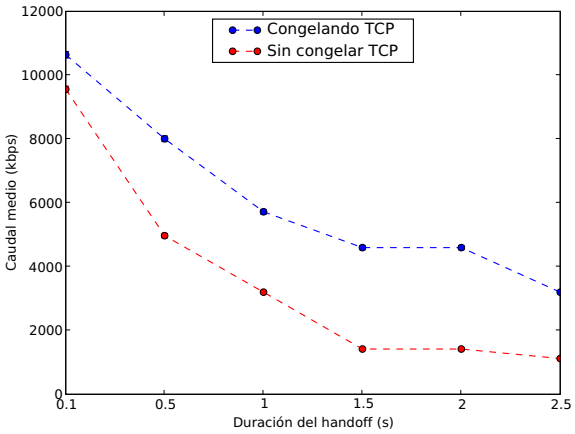
Con el objetivo de evaluar la mejora que introduce la congelación de los parámetros TCP en CRNs, se ha realizado un conjunto de simulaciones con el paquete de software ns-2 [9] cuyos resultados se han representado mediante las librerías matplotlib [10] de python. Todos los resultados que se presentan se han obtenido a partir de la media de 3000 iteraciones en simulaciones de 40 segundos.

El escenario de simulación consiste en una red 802.22 en la que se establece una conexión TCP entre dos usuarios secundarios. Como el sistema 802.22 especifica eficiencias espectrales en el rango de 0.5 bit/(s/Hz) a 5 bit/(s/Hz), considerando una media de 3 bits/(s/Hz), se ha dotado a la capa de red con una velocidad de transmisión de 18 Mbps (canal de televisión de 6 MHz) [11]. Asumiendo cabeceras IP de 20 bytes, la velocidad de transmisión a nivel de transporte es de 17.66 Mbps. Dado que el estándar 802.22 establece un rango de cobertura de la estación base de 33 Km para 4 W CPE EIRP, se ha asumido una distancia media de 15 Km entre cada uno de los secundarios y la estación base. Como la velocidad de propagación en el aire es de $3{\cdot}10^8 \frac{m}{s}$, el retardo de propagación definido entre cada secundario y la estación base es de $50\mu s$. Se considera despreciable el tiempo de proceso de tramas por parte de la estación base. Se asume también para la simulación una tasa de error de bit (*bit error rate* - BER) típica para 802.22 de 10^{-6} [12].

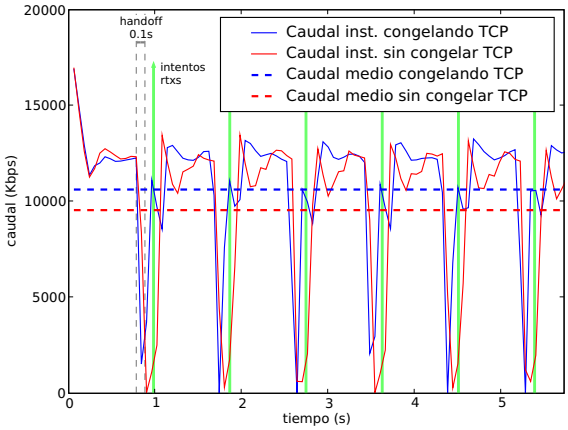
Se ha utilizado una fuente FTP que genera segmentos TCP de 1040 bytes con dos implementaciones de TCP: TCP Reno estándar y TCP Reno congelando parámetros cuando se produce un *handoff*. La única diferencia entre ambas implementaciones es que la segunda congela los parámetros de control de congestión, i.e, la ventana de congestión y el umbral, así como los temporizadores de retransmisión de los segmentos en tránsito (pendientes de confirmación) durante el *handoff*. De este modo, cuando éste finaliza, la transmisión se reanuda manteniendo el valor previo de dichos parámetros. Por el contrario, TCP Reno estándar es completamente ajeno a lo que ocurre a niveles inferiores y continua con la transmisión durante el *handoff*, con lo que si la duración del mismo es suficientemente larga expirarán los temporizadores de retransmisión de los segmentos en tránsito. Como ya se ha explicado en II esto puede implicar un largo periodo de inactividad de la fuente.

En los resultado mostrados en la figura 4, se asume un ataque Lion en el que el atacante sólo tiene capacidad para hacer *jamming* en un canal en un momento determinado y, por lo tanto, debe observar el medio para detectar en qué canal está operando la CRN víctima. El atacante busca hacer el mayor daño posible pero no tiene ninguna información acerca del canal al que conmutará la CRN después del *handoff*, es decir, que no sabe interpretar la información de control de la CRN o bien ésta está cifrada (contramedida propuesta en la sección III). Cuando el atacante detecta el nuevo canal de operación de la CRN vuelve a hacer *jamming* para provocar otro *handoff*. El tiempo que transcurre desde que la CRN reanuda la transmisión hasta que el atacante consigue forzar un nuevo *handoff* es en media de 781ms, valor que hemos considerando fijo para la simulación. Asumiendo una ocupación del 45 % de los canales de televisión, quedan 36 libres (en E.E.U.U las estaciones de TV operan en un total de 67 canales de las bandas UHF y VHF) para uso no licenciado. Con el objetivo de no interferir las señales primarias, se requieren 3 canales vacíos entre cada par de canales de TV para que la red 802.22 pueda operar [11]. Por tanto, esta reduce a 12 los canales disponibles. Suponiendo que el atacante debe observar en media la mitad de esos canales para determinar cuál esta siendo utilizado por la CRN y que el tiempo de observación para detectar su presencia es de 46.95ms [13], el tiempo total invertido por el atacante será de $6{\cdot}46{,}95\text{ms} = 281{,}7\text{ms}$. A partir de este momento, transcurrirán al menos 500ms [11] desde que el atacante realiza el PUE y la CRN detecta la presencia de un primario, hasta que ésta última deja de transmitir en esa banda frecuencial (*channel move time*). Por lo tanto, el tiempo mínimo que transcurrirá antes de que se produzca un nuevo *handoff* será de $500\text{ms} + 281{,}7\text{ms} = 781{,}7\text{ms}$.

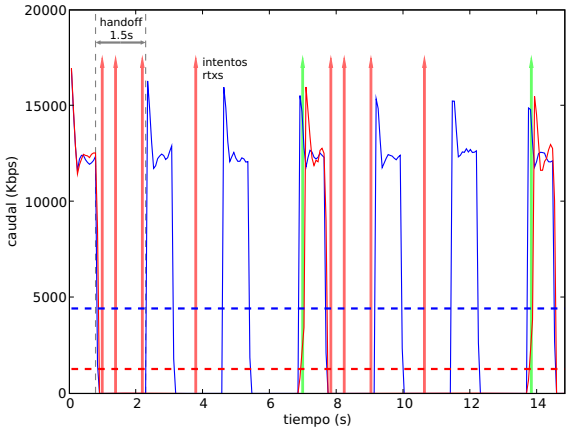
La figura 4a nos permite comparar el caudal medio TCP cuando se recibe el ataque mencionado con las dos implementaciones para diferentes valores de *handoff* (0.1s, 0.5s, 1s, 1.5s, 2s y 2.5s). Como se puede observar en la gráfica, congelando el TCP durante el *handoff*, el caudal TCP medio es muy superior al obtenido sin congelar; hecho que se acentúa a medida que se incrementa el tiempo de *handoff* necesario para que la CRN cambie de canal. Esta diferencia de caudal se justifica porque congelando TCP prácticamente se aprovecha todo el tiempo disponible para transmitir, ya que la recuperación de la comunicación es inmediata. Sin embargo, el TCP sin congelar continua transmitiendo segmentos durante el periodo de *handoff*, con lo que si es suficientemente grande, el temporizador de retransmisión de estos segmentos expirará. Hecho que implica una reducción del caudal TCP por dos razones: 1) la ventana de congestión se reduce a un sólo segmento; y 2) cada vez que se retransmite un segmento se dobla el temporizador de retransmisión. La primera, prácticamente no afecta en redes 802.22, dado que la ventana óptima de transmisión tiene un valor de aproximadamente un segmento (ver expresión 2). Sin embargo, la segunda puede provocar que TCP esté inactivo durante un largo periodo de tiempo.



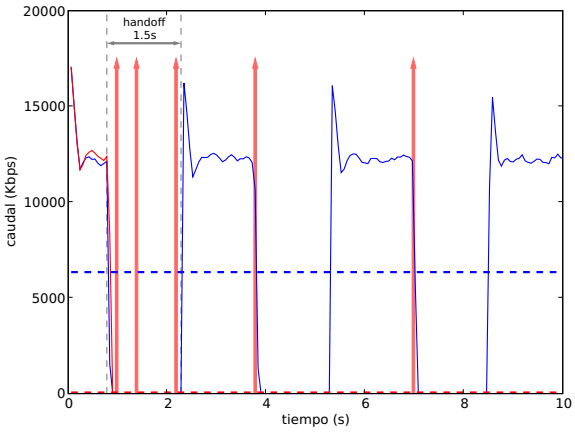
(a) Caudal vs. Handoff



(b) Handoffs de 0,5s



(c) Handoffs de 0,5s



(d) Ataque Lion inteligente (handoff=1,5s)

Figura 4. Efecto del ataque Lion sobre el caudal TCP congelando y sin congelar la transmisión

$$RTT = t_{tx} + 2t_{prop} \approx 641\mu s \quad (1)$$

$$W_{opt}(segmentos) = \frac{RTT}{t_{tx}} \approx 1{,}42 \quad (2)$$

En las figuras 4b y 4c mostramos el caudal TCP instantáneo y medio para valores de *handoff* de 0.1s y 1.5s respectivamente. Como se puede observar, y como ya se ha explicado en II, el intervalo de tiempo entre retransmisiones de un mismo segmento se dobla con cada intento fallido. El valor inicial del temporizador es de 200ms, que es un valor mínimo para la estimación del RTT, cuyo valor real es, en este caso, menor (ver expresión 2). Por lo tanto, si la transmisión de un segmento falla en t=0, los intentos de retransmisión se producirán en t=[0.2, 0.6, 1.4, 3, ...]s. En ambas figuras se ha marcado los instantes de retransmisión debidos al *handoff* como flechas verticales de color rojo, si todavía no hay enlace, y verdes si en ese instante se puede transmitir.

En la figura 4c se puede apreciar como el primer *handoff* se produce en el instante $t = 781$ms con una duración de 1.5s. La primera retransmisión se realiza 200ms después de la transmisión fallida, después de que se inicie el *handoff*, es decir, justo después del instante $t = 781$ms$+200$ms$= 981$ms, en el que el *handoff* todavía no ha finalizado. Las dos siguientes se producen justo después de $t = 981$ms$+400$ms$= 1381$ms y $t = 1381$ms$+800$ms$= 2181$ms, también durante el *handoff*. En el instante $t = 781$ms$+1500$ms$= 2181$ms termina el primer *handoff*, pero TCP permanece inactivo (esperando a retransmitir) hasta $t = 2181$ms$+1600$ms$= 3781$ms, que justamente coincide durante el segundo *handoff*. La siguiente retransmisión es ya en $t = 3781$ms$+3200$ms$= 6981$ms que coincide con un periodo de comunicación y, por tanto, tiene éxito. Como se puede observar, TCP Reno ha permanecido inactivo durante casi 7 segundos.

Por otro lado, en la figura 4b donde el periodo de *handoff* tiene un valor de 0.1s, la probabilidad de que los instantes de

retransmisión coincidan con un periodo de comunicación es mayor, con lo que el periodo de inactividad está limitado a un RTO y, por lo tanto, la diferencia de caudal medio entre las dos implementaciones TCP es mínima.

Hemos visto hasta ahora un ataque Lion poco inteligente en que el atacante sólo va detectando el canal de operación de la CRN y forzando un nuevo *handoff*, p.e. mediante un ataque PUE. Sin embargo versiones más "inteligentes" del ataque pueden llevar a reducir aún más el caudal TCP. Dado que el RTT asociado a las conexiones en CRNs es muy pequeño, la mayoría de implementaciones TCP utilizan un valor típico de RTO de 200ms que coincide con su estimación mínima del RTT. El valor del temporizador es por tanto conocido para el atacante. Mediante esta información, el atacante puede llegar a conseguir una DoS mediante un ataque Lion forzando *handoff*s que coincidan con los instantes de retransmisión de los segmentos. La figura 4d muestra un ejemplo de ataque Lion inteligente cuando el *handoff* es de 1.5s. Como se puede observar, el caudal TCP sin congelar parámetros se reduce a cero, mientras que congelando sólo se paraliza la transmisión durante los periodos de *handoff*.

V. Conclusion

Las redes cognitivas emergen como una solución prometedora para la escasez de espectro radioeléctrico. Estas redes están formadas por CRs que de forma inteligente deciden cuáles son las mejores oportunidades de espectro. A pesar de que las CRNs se basan en tecnologías existentes, para proporcionar seguridad en este tipo de redes no basta con aunar los mecanismos que se aplican en dichas tecnologías. Como consecuencia de las particularidades de las CRNs, aparecen nuevos ataques y algunos de los que ya existían aumentan en complejidad. Por este motivo, se requieren nuevas propuestas para responder eficientemente ante estos ataques específicos, en particular en las capas física y de enlace. Además, es necesario un mecanismo exhaustivo que permita prevenir ataques en todos los niveles de la torre de protocolos.

En este documento se presenta un nuevo ataque *cross-layer* a CRNs denominado ataque Lion. Este ataque se basa en realizar *jamming* en la banda frecuencial en la que opera una CRN determinada forzándola a realizar un cambio de frecuencia o *handoff*. La interrupción de las conexiones debido al *handoff* provoca pérdidas y aumenta artificialmente el *round-trip-time* de las mismas, hecho que conlleva la degradación del caudal de TCP e incluso en algunos casos, una reducción total.

Con el objetivo de mitigar el ataque se ha propuesto un conjunto de medidas que abarcan desde la criptografía aplicada hasta mecanismos de protección *cross-layer*. Estas potenciales soluciones son complementarias y por tanto deben ser aplicadas simultáneamente para minimizar el impacto del ataque. En primer lugar, se ha sugerido algunas modificaciones en el protocolo TCP para evitar la degradación del rendimiento debido a los *handoffs*. Dichos cambios implican el intercambio de información entre las capas física/enlace y la de transporte. De este modo, los miembros de la CRN pueden congelar los parámetros de las conexiones TCP durante el *handoff* y adaptarlos a las nuevas condiciones, ya conocidas antes de que se produzca. En segundo lugar, se ha considerado la necesidad de dotar de seguridad a la información de control para evitar que un posible atacante pueda espiar las acciones de la CRN presentes y futuras. Con este objetivo, se ha propuesto el uso de técnicas de gestión de claves de grupo (GKM), que proveen de un sistema muy eficiente de gestión de la seguridad en grupos dinámicos de usuarios, como es el caso de muchas CRNs. Por último, se ha propuesto el uso de sistemas de detección de intrusos (IDSs) específicamente adaptados a CRNs. A pesar de que tradicionalmente los IDSs son centralizados y dirigidos a una única capa, se ha justificado el uso de técnicas *cross-layer* en estos sistemas y la necesidad de cooperación para poder detectar amenazas en CRNs de forma efectiva.

Con el objetivo de evaluar el impacto del ataque Lion sobre las conexiones TCP, se ha realizado un conjunto de simulaciones con y sin la modificación propuesta de congelar los parámetros de transmisión TCP. Los resultados obtenidos demuestran que la congelación de parámetros permite reducir el efecto de los *handoffs* (provocados por el ataque) sobre el caudal de TCP. Además, en el caso que el atacante provoque de forma "inteligente" los *handoffs*, evita que el ataque se convierta en una DoS.

Con el auge de las CRNs, los requisitos de seguridad aumentarán en este tipo de redes. Se espera que el trabajo que se está desarrollando en las áreas de privacidad y confianza convierta a las CRNs en una opción especialmente atractiva en un amplio conjunto de escenarios.

Agradecimientos

Este trabajo ha sido posible gracias a los proyectos "P2Psec" (CICYT - TEC2008-06663-C03-01) y CONSOLIDER CSD2007-00004 "ARES".

Referencias

[1] R. Chen and J.-M. Park, "Ensuring trustworthy spectrum sensing in cognitive radio networks," in *1st IEEE Workshop on Networking Technologies for Software Defined Radio Networks (SDR)*, Sep. 2006, pp. 110–119.

[2] T. Clancy and N. Goergen, "Security in cognitive radio networks: Threats and mitigation," in *3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom)*, May 2008, pp. 1–8.

[3] I. Aad, J.-P. Hubaux, and E. W. Knightly, "Denial of service resilience in ad hoc networks," in *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2004, pp. 202–215.

[4] D. Wallner, E. Harder, and R. Agee, "Key management for multicast: issues and architectures," RFC 2627, 1998.

[5] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2000, pp. 275–283.

[6] A. Mishra, K.Ñadkarni, and A. Patcha, "Intrusion detection in wireless ad hoc networks," *Wireless Communications, IEEE*, vol. 11, no. 1, pp. 48–60, Feb 2004.

[7] V. Bhuse and A. Gupta, "Anomaly intrusion detection in wireless sensor networks," *Journal of High Speed Networks*, vol. 15, no. 1, pp. 33–51, 2006.

[8] M. Mejia, N. Pena, J. L. Munoz, and O. Esparza, "A review of trust modeling in ad hoc networks," *Internet Research*, vol. 19, no. 1, pp. 88–104, 2009.

[9] "The network simulator - ns-2." [Online]. Available: http://www.isi.edu/nsnam/ns/
[10] "Matplotlib." [Online]. Available: http://matplotlib.sourceforge.net/
[11] C. Cordeiro, K. Challapali, D. Birru, and N. Shankar, Sai, "Ieee 802.22: an introduction to the first wireless standard based on cognitive radios," *Journal of Communications*, vol. 1, no. 1, pp. 38–47, Apr. 2006.
[12] C. Stevenson, C. Cordeiro, and G. Chouinard, "Functional requirements for the 802.22 wran standard," Jan. 2006.
[13] G. Chouinard, D. Cabric, and M. Gosh, "Ieee p802.22 wireless rans - sensing thresholds," May 2006. [Online]. Available: https://mentor.ieee.org/802.22/dcn/06/22-06-0051-04-0000-sensing-thresholds.xls

Distributed Evolution of Strategies in a Game Theoretic Trust Model for Mobile Ad Hoc Networks

Marcela Mejia[1], Marco Alzate[2], Jose L. Muñoz[3], Néstor Peña[1] and Oscar Esparza[3]

[1]Universidad de los Andes
[2]Universidad Distrital
[3]Universidad Politécnica de Cataluña

am.mejia75@uniandes.edu.co, malzate@udistrital.edu.co,
jose.munoz@entel.upc.edu, npena@uniandes.edu.co, oscar.esparza@entel.upc.edu

***Abstract*—Cooperation among nodes is fundamental for the operation of mobile ad hoc networks (MANETs). However, in these networks there could be selfish nodes that use resources from other nodes to send their packets but do not offer their resources to forward packets for other nodes. Several trust models have been proposed as mechanisms to incentive cooperation in MANETs. Some of them are based on game theory concepts. Among game theoretic trust models, those that make nodes' strategies evolve genetically have shown promising results for cooperation improvement. However, current approaches propose a highly centralized genetic evolution so they cannot properly adapt to fast changing conditions. In this paper, we propose a game theoretic trust model that uses a bacterial-like algorithm to let the nodes quickly learn the appropriate cooperation behavior. Our model is completely distributed and achieves good cooperation values in a small fraction of the time compared with centralized algorithms.**

***Keyword*—MANET, Trust models, Game theory, evolutionary algorithm**

I. Introduction

Mobile Ad Hoc NETworks (MANETs) are infrastructureless networks formed by wireless mobile devices with limited resources. Source/destination pairs that are not within transmission range of each other must use intermediate nodes as relays [1]. MANETs are particularly vulnerable to selfish behavior, as some nodes may prefer saving resources instead of forwarding packets on behalf of others [2]. Thus, it is important to encourage the nodes to participate in essential network functions such as packet routing and forwarding. In this sense, several trust models have been proposed as mechanisms to incentive node participation within the network [3]. A trust model is a conceptual abstraction to build mechanisms for assigning, updating and using trust levels between the entities in a distributed system [4]. The trust model becomes a tool for helping the subject of a distributed system to select the most reliable agent among several others offering a service [4]. Many different mechanisms have been used in the literature to design trust models for mobile ad hoc networks [3]. The problem of deciding whether to cooperate (and improve the trust) or not to cooperate (and save resources) can be seen as a game. For this reason, many of the proposals in the literature are based on game theory concepts [2], [5], [6]. Among the proposals in the literature, the trust model presented by Seredynski et. al. in [7] is interesting because it presents a way of dynamically adapting the collaboration strategy to the network conditions. The evolution is performed using a genetic algorithm that presents promising results regarding cooperation improvement. However, there are still serious concerns about the highly centralized nature of the approach and its slow convergence. Indeed, the optimal strategies are obtained by using a centralized entity that runs a conventional genetic algorithm after a large number of interactions between nodes to evolve the set of strategies. Taking into account the drawbacks of the previous proposal, in this paper we present a distributed bacterial-like evolution algorithm based on a few interactions among nodes. Our model does not assume a central entity nor requires an unrealistically large number of interactions among nodes to evolve the strategies. It is based on distributed parallel cellular genetic algorithms [8],[9],[10],[11],[12] where genetic information is interchanged among neighbor nodes (much like plasmid migration in bacterial colonies [13],[14]). This way, each individual node selects the strategy that locally maximizes its payoff in terms of packet delivery and resource saving. This local payoff maximization is such that globally the whole network increases the cooperation (and, consequently, the throughput) and reduces the resources wasted serving selfish nodes. The rest of the paper is organized as follows. Section II briefly describes the trust model of [7], because we base our proposal on it. Section III introduces our trust model and its evolutionary algorithm. Section IV shows some numerical results and compares them with the previously published results of [7] and with a theoretical upper bound on the maximum achievable cooperation. Section V concludes the paper.

II. Centralized Evolution Algorithm

In [7], the interactions among nodes are based on the iterated prisoner's dilemma under the random pairing game [15]. Each intermediate node utilizes a strategy that defines whether it should retransmit or discard a packet that comes from a certain source node. The strategy depends on two aspects: the past behavior of the network when the intermediate node acted as source node and the trust level that the intermediate

node has in the source node. The model is comprised of a trust evaluation mechanism, a game based network model, a strategy and a genetic algorithm to evolve the strategy.

A. Trust evaluation mechanism

Each node maintains a trust table based on the observed behavior of its neighbors. For example, if node B is observing node A, which is within its transmission range, it can know the number of packets that has been sent to A to be forwarded, pcs_A, and the number of packets that A has actually forwarded, pcf_A. So B can compute the forwarding rate of A $f_r(B, A) = pcf_A/pcs_A$. With this rate, B can determine the trust level it should have in A, $T\{B : A\}$, as shown in Table I.

Tabla I
RELATION BETWEEN DELIVERY RATE AND TRUST LEVEL

$f_r(B, A)$	$T\{B : A\}$
$1 - 0.9$	3
$0.9 - 0.6$	2
$0.6 - 0.3$	1
$0.3 - 0$	0

B. Game-based network model

Each game starts with the transmission of a new packet from a source node and ends either when the packet is delivered to its destination, or when an intermediate node decides to discard the packet. Once the game has finished, each participant receives a payoff according to the decision it took and its trust level on the source node. In this model, two types of nodes are defined: source nodes and intermediate nodes. Therefore, two types of payoff tables are maintained, as shown in Table II.

Tabla II
PAYOFF TABLES

Source Node	
Transmission Status	Payoff
Successful	5
Failed	0

Intermediate Node				
	Trust Level			
Decision	$T = 3$	$T = 2$	$T = 1$	$T = 0$
Cooperate	3	2	1	0.5
Discard	0.5	1	2	3

C. The strategy

The strategy that a node has to follow when it is acting as intermediate node is represented by a string of bits. Each bit represents a decision (Discard (0)/ Cooperate (1)) taking into account a set of parameters. In [7], this set of parameters is formed by: (i) the transmission status of the two previous games that the node has played as source (which could be success (S) or failure (F)) and, (ii) the trust level that the node has in the source node. The resulting strategy has 18 bits and an example is shown in TableIII. Notice that two additional bits are used when the node has played less than two games as source node.

Tabla III
STRATEGY CODING, EXAMPLE STRATEGY 0000 0011 0101 1010 11

Source Trust Level	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3	
Transmission Status −2	S	F	S	F	S	F	S	F	S	F	S	F	S	F	S	F	
Transmission Status −1	S	S	F	F	S	S	F	F	S	S	F	F	S	S	F	F	
Current Decision	D	D	D	D	D	D	C	C	D	C	D	C	C	D	C	D	C	C

D. The genetic algorithm

The genetic algorithm aims to maximize the mean payoff of each node. Initially, nodes have a randomly generated strategy and, then, series of "tournaments" are played to calculate the payoff that nodes will receive for their actions. The fitness of each player's strategy is evaluated as the average payoff per event. According to this fitness, $2N$ strategies are selected through a roulette wheel mechanism, where N is the number of participating normal nodes. Applying a standard one point crossover and a standard uniform bit flip mutation over these $2N$ strategies, a set of N new strategies is generated and the whole process is repeated during a certain number of generations. The results of the simulations in [7] show that the strategies evolve to adapt to different environments, where an environment is characterized by a given number of selfish and normal nodes. Finally, we clarify some definitions used in [7]. A "tournament" is played among 50 nodes, randomly selected from a total population of 100 nodes, where each tournament is composed of 300 "rounds". A round is composed of 50 (successful or failed) packet transmissions or "games". Since each of the 100 nodes must participate in at least two tournaments per generation, the average number of tournaments per generation is 11.56, for a total of $11.56 \times 300 \times 50 \div 100 = 1734$ packets per node per generation, in average. This number will be important to compare the efficiency of this centralized algorithm with our distributed algorithm.

III. DISTRIBUTED BACTERIAL-LIKE EVOLUTION ALGORITHM

The proposal in [7] fairly represents the dilemma of forwarding packets to gain trust or discard them to save energy in a MANET. However, it has an expensive fitness evaluation mechanism and a highly centralized evolution algorithm, since, after each tournament, a central entity should collect all the strategies and their firtness in order to compute and redistribute the new evolved strategies. This makes the model unfeasible for implementation in a real MANET. For this reason, we propose and evaluate a distributed model based in [7], in which we keep the trust evaluation mechanism, the game based network model and the strategy but change the genetic algorithm to evolve the strategy. Indeed, we omit both the central entity and the costly process of having multiple generations by allowing the nodes to exchange genetic material among their neighbors, like plasmid migration in bacterial colonies [13]. A plasmid is an extrachromosomal DNA molecule that bacteria can take up from the external environment, in order to obtain a gene that gives the cell a

selective advantage. When the cell replicates, it makes copies of the acquired plasmid [14]. This model can be used in evolutionary algorithms instead of the traditional Darwinist method used in [7]. The plasmid migration is a greedy algorithm that, at each step, makes apparent good decisions without regarding for future consequences and, as such, can lead only to locally optimal solutions. In contrast, these solutions can be obtained very quickly, enhancing adaptability at the cost of optimality. For us, the foremost characteristics of the algorithm are its distributed implementation and its convergence speed. These features make the algorithm readily implementable in a MANET environment.

In our algorithm, we evolve the strategies on-line during the life of the network instead of using a centralized entity to run the genetic algorithm for each generation, and different tournaments to evaluate the strategies at each generation. So, we keep the concepts of "game" and "round" of [7] but omit those of "tournament" and "generation". A game is a successful or failed packet transmission for which a source node selects the most trusted h-hop route among r possible routes. The number of hops, h, obeys a probability distribution p_h and, given h, and the number of routes, r, obeys a conditional probability distribution $p_{r|h}$. These distributions will be used below to compute the maximum achievable cooperation. A round is a set of 100 games in which each node plays once a game as source. The evolution goes through periods of R rounds, called *Plasmid Migration Periods* (PMP), after which every node interchanges genetic information with its neighbors. Figure1 shows how our distributed model works. Each node starts with a random strategy, whose fitness is evaluated during a PMP of R rounds. At the j^{th} PMP, node i interchanges its strategy $s_i(j)$ and its fitness $f_i(j)$ with its one-hop neighbors. Each node selects a potential parent strategy among the neighbor strategies through the roulette wheel process, and then:

- If the current fitness $f_i(j)$ of the node is better than the selected one $f_N(j)$, the node keeps its own strategy by skipping the crossover process.
- Otherwise, the selected strategy is combined with the current strategy of the node using a one point crossover.

Finally, the resultant strategy suffers a standard uniform bit flip mutation process before going to a new PMP for evaluation. The process is repeated during the life time of the network. If we choose the number of rounds in a PMP as R = 1734, a PMP in our distributed algorithm would be equivalent to a generation in the centralized algorithm of [7]. However, our algorithm converges much faster, which allows us to use lower values for R and, at the same time, have a convergence in a PMP similar to the convergence achieved in a generation in [7]. This means that our algorithm will be close to $1734/R$ times faster than the centralized one. In the next section we perform some experiments to determine R.

IV. Numerical Results

In this section we show some evaluation results to demonstrate the usability of our proposal for the MANET scenario. As in [7], all the simulations have been independently replicated 60 times. One of the first experiments was aimed at deciding R (i.e. the length of a PMP). For this purpose, we

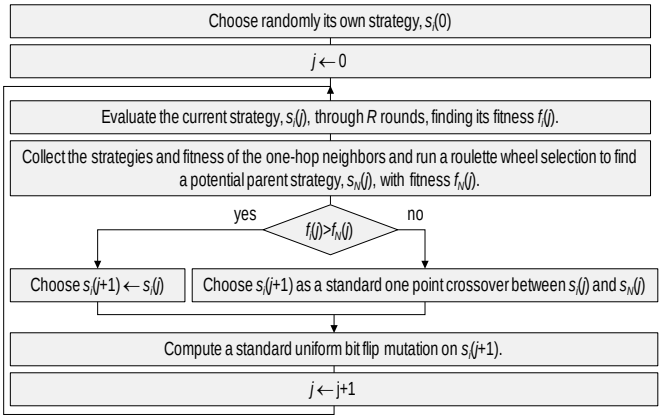


Fig. 1. On-Line distributed evolutionary algorithm

computed the cooperation evolution for R = 25, 75 and 300 rounds under different number of selfish nodes within the population of 100 nodes. The results obtained (not included here due to space constraints) show that in all the cases the cooperation converged to steady values after a few hundred plasmid migration periods. The cooperation achieved with 75 and 300 rounds per migration period was very similar in every scenario, although the convergence was faster with 75 rounds per PMP than with 300. With 25 rounds per PMP, not only the convergence took longer, but also converged to lower cooperation values. Because of these results, we decided to use a PMP of R = 75 rounds. Figure 2 plots the average cooperation as a function of the number of packets generated per node, comparing the evolution speed of our distributed algorithm and the centralized algorithm of [7] under different fraction of selfish nodes. We compare the first 250 generations of the centralized algorithm, corresponding to an average of 433500 packets generated per node (in grey color), and the first 2500 PMPs of our distributed algorithm, corresponding to 187500 packets generated per node (in black color). The continuous lines correspond to a scenario with no selfish nodes, the dashed lines correspond to a scenario where 20% of the nodes are selfish, the dashed-dotted line is for 50% of selfish nodes and the dotted line is for 60% of selfish nodes. The centralized algorithm starts at 1734 packets transmitted per node because that corresponds to the first generation. The distributed algorithm starts at 75 packets, corresponding to the first PMP.

The figure2 shows that our algorithm converges much faster than the centralized one. The improvement achieved over the convergence speed is above an order of magnitude, although the best cooperation achieved is lower for our distributed algorithm than for the centralized one when there are selfish nodes within the network. This behavior can be explained because (1) the parallel evolution only considers the genetic information of local individuals instead of the global information of the whole population, (2) the fitness is not as thoroughly evaluated with 75 transmitted packets as with 1734, and, fundamentally, (3) the nodes carry the reputation of the strategies they used before the previous PMP.

In order to compare the maximum achieved cooperation of the centralized and distributed algorithms, below we develop a theoretical expression of the maximum achievable cooperation

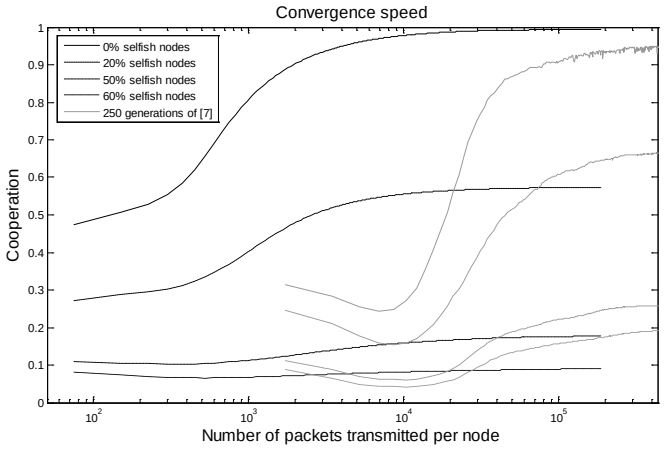


Fig. 2. Maximum achievable cooperation values

under an optimal strategy with perfectly computed trusts. For this purpose, we assume that the selfish nodes are completely identified and that, with this information, the normal nodes cooperate among them and discard the packets of selfish nodes. This is the ideal condition that any trust model would like to achieve. In this condition, a packet will get through whenever the path is composed exclusively of normal nodes, which occur with the probability shown in Equation (1), where P_h is the probability that the path length is h hops, $P_{r|h}$ is the probability of finding r routes given that the path length is h hops, A is the event *a packet finds at least one route made out exclusively of normal nodes* and B(h) is the event *there are no selfish nodes among h randomly selected nodes*. Clearly, the ideal condition implies that the best possible cooperation, C_{best}, is the probability of A. The probability of $B(h)$ is given in Equation (2), where N_N is the number of normal (not selfish) nodes and N is the total number of nodes.

$$C_{best} = P_r[A] = \sum_h \sum_r P_h P_{r|h}(1 - (1 - P_r[B(h)])^r) \quad (1)$$

$$P_r[B(h)] = \prod_{i=0}^{h-1} \frac{N_N - i}{N - i} \quad (2)$$

Figure3 compares the theoretically maximum achievable cooperation with the maximum values obtained through the centralized and distributed algorithms. Except when there are no selfish nodes, our best values are lower than those of the centralized algorithm and even farther from the optimal ones. Nevertheless, our values, which are obtained on-line in a distributed way, are close to those obtained in a centralized way.

There is a tradeoff between optimality and adaptability in terms of the length of the PMP, since it will determine the accuracy of the fitness evaluation. In [7], since each node generates an average of 1734 packets per generation before going through the genetic evolution algorithm, there is a highly accurate estimation of the fitness at the cost of a prohibitively large convergence time. Besides, in [7] the central entity knows the strategies of all the nodes and their fitness, so this central entity knows the entire space of feasible strategies to compute the next generation and distribute it over all the nodes. In our approach, we accept a higher variance in the evaluation of the fitness by reducing the PMP to only a few node interactions. This allows us both to replace the central entity by a distributed plasmid migration based on locally interchanged information over one-hop neighbors, and to run this plasmid migration often enough for the nodes to converge on line to good strategies. This makes our scheme implementable in a real MANET, because it exploits its distributed organization and takes advantage of the clustered nature of its topology.

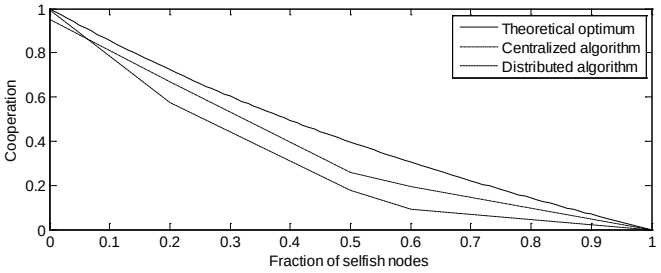


Fig. 3. Maximum achievable cooperation values

V. Conclusions

In this paper we have shown that it is possible to use distributed algorithms for the genetic evolution of strategies in a game theoretic trust model for MANETs. We have proposed a trust model in which nodes interchange genetic information among neighbor nodes, much like plasmid migration in bacterial colonies. This way, each node adapts its strategy to the dynamical characteristics of the network, maximizing its payoff in terms of packet delivery and resource saving. Our proposal does not need a central entity and does not require unrealistically large number of node interactions to evaluate the fitness. The numerical results show that our algorithm can quickly find good strategies, more than 20 times faster than the corresponding centralized algorithm. Further work will modify the trust computation to reflect the change of strategies at each PMP and will evaluate the adaptability to changing conditions.

Acknowledgements

This work was partly supported by the Colombian Institute for science and Technology development, Colciencias, Universidad Nueva Granada and Universidad de los Andes, in Colombia, as well as the Spanish Government through projects CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES", TSI2007-65393-C02-02 "ITACA" and TSI2005-07293-C02-01 "SECONNET", and by the Government of Catalonia under grant 2005 SGR 01015 to consolidated research groups.

References

[1] Charles E. Perkins. *Ad Hoc Networking*. Addison-Wesley Professional, 1 2001.

[2] Konrad Wrona and Petri Mähönen. Analytical model of cooperation in ad hoc networks. *Telecommunication Systems*, Volume 27:347 – 369, 2004.

[3] Marcela Mejia, Néstor Peña, José. L. Muñoz, and Oscar. Esparza. A review of trust modeling in ad hoc networks. *Internet Research*, volume 19-1(1):88–104, 2009.

[4] Sergio Marti and Hector Garcia-Molina. Taxonomy of trust: categorizing p2p reputation systems. *Comput. Netw.*, 50(4):472–484, 2006.

[5] Juan José Jaramillo and R. Srikant. Darwin: distributed and adaptive reputation mechanism for wireless ad-hoc networks. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 87–98, New York, NY, USA, 2007. ACM.

[6] Lu Yan and Stephen Hailes. Cooperative packet relaying model for wireless ad hoc networks. In *FOWANC '08: Proceeding of the 1st ACM international workshop on Foundations of wireless ad hoc and sensor networking and computing*, pages 93–100, New York, NY, USA, 2008. ACM.

[7] M. Seredynski, P. Bouvry, and M.A. Klopotek. Modelling the evolution of cooperative behavior in ad hoc networks using a game based model. *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on*, pages 96–103, April 2007.

[8] Enrique Alba and BernabÃ© Dorronsoro. *Cellular Genetic Algorithms (Operations Research/Computer Science Interfaces Series)*. Springer, 1 edition, 6 2008.

[9] Enrique Alba and José M. Troya. A survey of parallel distributed genetic algorithms. *Complex.*, 4(4):31–52, 1999.

[10] Erick Cantpaz. A survey of parallel genetic algorithms. *Calculateurs Paralleles*, 10, 1998.

[11] M. Nowostawski and R. Poli. Parallel genetic algorithm taxonomy. *Knowledge-Based Intelligent Information Engineering Systems, 1999. Third International Conference*, pages 88–92, Dec 1999.

[12] Enrique Alba and Bernabé Dorronsoro. Auto-adaptación en algoritmos evolutivos celulares, un nuevo enfoque algoritmico. In *II Congreso Español sobre Metahurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB03)*, 2003.

[13] I. W. Marshall and C. Roadknight. Adaptive management of an active service network. *BT Technology Journal*, 18(4):78–84, 2000.

[14] Jeremy Dale and Simon Park. *Molecular Genetics of Bacteria*. Wiley, 4 edition, 3 2004.

[15] H. Ishibuchi and N. Namikawa. Evolution of cooperative behavior in the iterated prisoner's dilemma under random pairing in game playing. *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, 3:2637–2644 Vol. 3, Sept. 2005.

Mejora del encaminamiento en redes multi-salto con la SNR

Ramón Agüero, José Antonio Galache, Luis Muñoz
Departamento de Ingeniería de Comunicaciones - Universidad de Cantabria
Laborotarios I+D Telecomunicaciones. Plaza de la Ciencia - Santander
{ramon, jgalache, luis}@tlmat.unican.es

***Resumen*—Este artículo analiza la utilización de una métrica, basada en la relación señal a ruido (SNR) de los enlaces inalámbricos, para mejorar las prestaciones de los protocolos de encaminamiento sobre redes inalámbricas multi-salto. Para ello, compara el comportamiento y el rendimiento de un protocolo basado en dicha métrica con otras estrategias, como la más tradicional (basada en minimizar el número de saltos) u otras (como la basada en la *Expected Transmission Count*), que han suscitado un gran interés por parte de la comunidad científica. Para este estudio, se usa un análisis basado en técnicas de simulación, pero empleando un modelo de canal realista, capaz de emular adecuadamente el efecto *Gray Zones* que puede tener una influencia negativa en las comunicaciones inalámbricas, en general, y sobre topologías multi-salto, en particular. Los resultados ponen de manifiesto que es posible llevar a cabo mejoras relevantes utilizando la métrica propuesta para mejorar los procedimientos de encaminamiento sobre redes multi-salto.**

I. Introducción

Posiblemente, una de las áreas que más atención ha suscitado recientemente por parte de la comunidad científica es la conocida como redes *ad-hoc*, principalmente la problemática asociada al encaminamiento sobre este tipo de redes. Originalmente, este tipo de despliegues de red aparecieron como alternativa para posibilitar las comunicaciones en situaciones muy concretas en las que la presencia de una infraestructura no estuviese garantizada. Así, se postulaban en escenarios en los que las comunicaciones se originaran de manera espontánea, además de estar caracterizadas por su temporalidad. Por otro lado el concepto de red mallada (*mesh network*), ha aparecido recientemente como un nuevo modelo de comunicación, actualmente en desarrollo en diferentes grupos de trabajo del IEEE, en el entorno de las redes de área personal y local inalámbricas (WPAN, WLAN), 802.15 y 802.11, respectivamente. Además están siendo considerados en la especificación de la tecnología WiMax (IEEE 802.16).

Con el advenimiento del concepto de red personal, las topologías multi-salto jugarán definitivamente un papel muy relevante, ya que facilitarán la aparición de nuevos modelos de negocio, posibilitando, por ejemplo, que los operadores incrementen su portfolio de servicios y productos. A pesar de que el trabajo en el ámbito de las redes *ad hoc* comenzara en 1998, hay aún un gran abanico de aspectos técnicos sobre los que es necesario investigar. En este sentido, sería relevante destacar el hecho de que los protocolos de encaminamiento que habitualmente se han postulado para ser empleados sobre este tipo de topologías y que han gozado de una mayor relevancia, no consideran habitualmente aspectos de *calidad de servicio*. De esta forma, en la mayoría de los casos la selección de ruta se realiza en función del criterio de minimizar el número de saltos, que podría no ser la mejor elección bajo ciertas condiciones concretas, llevando a situaciones subóptimas, al considerar rendimiento, tasas de error, etc. En este trabajo, se demuestra que incrementar el número de métricas en el proceso de selección de ruta puede aportar beneficios relevantes para el comportamiento de la red, ya que de ese modo se pueden aliviar las hostilidades que presentan los entornos de propagación inalámbricos. Más concretamente, hace uso de las técnicas de optimización cruzada entre capas (*cross layer optimization*), utilizando una estimación de la calidad del enlace radio a través de su relación señal a ruido (SNR), para mejorar el algoritmo de selección de ruta empleado por el protocolo de encaminamiento *Dynamic Source Routing* (DSR) [1].

Aunque recientemente haya habido otros trabajos que han introducido métricas adicionales en el proceso de selección de rutas, ninguno de ellos analiza el beneficio de emplear la SNR. Además, y para asegurar un análisis realista, se hará uso de un modelo de error novedoso, basado en un filtrado auto-regresivo, que es capaz de reflejar el comportamiento *a ráfagas* y *con memoria* que caracteriza los entornos de propagación reales.

El artículo se estructura tal y como sigue: la Sección II presenta los trabajos existentes en la literatura relacionados con la temática. La Sección III presenta cuál ha sido el enfoque que se ha seguido en el algoritmo que se presenta en este trabajo, mientras que la Sección IV describe el modelo de canal que se ha empleado para evaluar el protocolo propuesto, capaz de reflejar, de manera realista, las condiciones de propagación en interiores. La Sección V plasma los resultados obtenidos en el rendimiento del algoritmo propuesto, comparándolo con otras estrategias presentes en la literatura. Finalmente, la Sección VI concluye el artículo, proponiendo algunos puntos que quedan abiertos para trabajos futuros.

II. Trabajo relacionado

Como se ha mencionado anteriormente, es posible encontrar, en la literatura relacionada, trabajos que han propuesto diferentes métricas para mejorar los algoritmos actuales de selección de ruta sobre topologías multi-salto. De Couto *et al.* [2] ya postularon que el enfoque tradicional basado en minimizar el número de saltos que sigue, por ejemplo, el protocolo DSR original, podría no ofrecer un rendimiento óptimo. En un trabajo posterior [3] proponen una métrica novedosa, la aceptada *Expected Transmission Count* (ETX), que tiene en cuenta la probabilidad de que la transmisión de una trama sobre un enlace determinado sea exitosa. Los autores demuestran, sobre una plataforma real, que el comportamiento del ETX mejora las estrategias de encaminamiento

tradicionales, aunque hay que decir que la tasa de transmisión se fijaba a 1 *Mbps* durante todos los experimentos.

A pesar de que existen otras aproximaciones al problema, por ejemplo Adya *et al.* [4] proponen utilizar el tiempo de ida y vuelta (*Round Trip Time*, RTT) para asignar las calidades correspondientes a cada enlace, mientras que Dube *et al.* [5] utilizan la potencia de señal recibida (pero no la SNR) para filtrar enlaces potencialmente temporales, la métrica ETX es sin ninguna duda la que ha acaparado mayor interés por parte de la comunidad científica. Draves *et al.* [6] la comparan con otras estrategias de encaminamiento, utilizando una plataforma real (con el mecanismo de selección de velocidad activado en las interfaces inalámbricas), concluyendo que ofrece un comportamiento más adecuado (sobre redes estáticas). La ETX también se ha utilizado como punto de partida para elaborar otras métricas; por ejemplo, Draves *et al.* [7] propusieron la *Expected Transmission Time* (ETT), una evolución de la ETX para ser empleada sobre redes heterogéneas. Además, Koksal y Balakrishnan [8] desarrollaron una versión modificada de la ETX (mETX), que trataba de solventar el problema de su predecesora, en tanto en cuanto no era capaz de reflejar adecuadamente las variaciones rápidas del canal inalámbrico. También proponen la *Effective Number of Transmissions* (ENT), que considera la influencia que una ráfaga de tramas perdidas podría tener sobre el comportamiento de los protocolos de capas superiores, como TCP. Como último ejemplo ilustrativo, se podría destacar el trabajo de Awerbuch *et al.* [9], que define la *Medium Time Metric* (MTM), que busca la selección de caminos de alto rendimiento, evitando aquellos enlaces que pudieran resultar poco fiables.

Por otro lado, alguno de los trabajos anteriormente mencionados, por ejemplo [8], [9], ya han puesto de manifiesto alguna de las desventajas que presenta la métrica ETX. Posiblemente la más relevante es que se basa en la difusión (modo *broadcast*) de mensajes sonda (*probe*) para estimar la calidad de los enlaces inalámbricos; la transmisión *broadcast* se realiza a una velocidad binaria menor que la de trabajo (en IEEE 802.11b a 2 *Mbps*), lo que se deriva en una cobertura sensiblemente mayor que la correspondiente a transmisiones en modo *unicast*. Sin embargo, es importante destacar que en los trabajos originales de De Couto *et al.* [3] y Draves *et al.* [6], la diferencia podría no ser tan relevante, teniendo en cuenta la configuración concreta de las tarjetas inalámbricas durante los experimentos. Además, el efecto descrito con anterioridad (diferencias entre las coberturas en función del régimen binario), que ha recibido el nombre de *Gray Zones* y que fue originalmente descrito por Lundgren *et al.* [10], tiene una influencia muy adversa en la estimación de la ETX, tal y como Kim *et al.* [11] ya han demostrado.

Un denominador común para la mayoría de los trabajos anteriores es que se basan en una validación empírica. Es evidente, y no admite discusión, el hecho de que ese tipo de aproximaciones aseguran un claro valor añadido a todo el trabajo, aunque, por otro lado, también presenta alguna limitación, ya que no facilita una evaluación sistemática y profunda de los diferentes algoritmos; mientras que además introduce ciertas dificultades al incorporar, por ejemplo, movilidad en los nodos, dentro de los análisis. Por tanto, una evaluación basada en técnicas de simulación incorpora ciertas ventajas en lo que se refiere a su capacidad para analizar más adecuadamente el comportamiento de las diferentes propuestas. De hecho, Koksal y Balakrishnan [8] utilizan trazas reales (obtenidas sobre canales inalámbricos) para realizar evaluaciones exhaustivas de las métricas que proponen, aunque no pueden llegar a la riqueza que una simulación real puede ofrecer en ese sentido.

Por tanto, se puede decir que este trabajo tiene un claro valor añadido; por un lado ofrece una validación extensa de diferentes métricas, utilizando un modelo de canal realista (pero no trazas concretas), capaz de reflejar adecuadamente el comportamiento observado empíricamente sobre entornos de propagación en un escenario típico de oficinas. Además, el uso de la SNR para estimar la calidad de los enlaces correspondientes es un aspecto que, por ejemplo, Koksal y Balakrishnan [8] destacaban como aspecto interesante a acometer en futuros trabajos.

III. SNR Aware DSR

Al igual que otros trabajos similares [3], [6], el protocolo *SNR Aware DSR* (SADSR) se basa en la operación básica de DSR, ya que su enfoque basado en encaminamiento fuente es un aspecto fundamental a la hora de potenciar el uso de métricas adicionales en los algoritmos de selección de rutas. A continuación se describen los aspectos fundamentales de su operación.

A. Evolución del DSR al SADSR

El cambio fundamental que es necesario acometer es el de asignar pesos apropiados a cada uno de los enlaces inalámbricos, de acuerdo a la calidad percibida. Posteriormente, un algoritmo de búsqueda de camino de coste mínimo (típicamente *Dijkstra*) se puede usar para encontrar el camino óptimo entre el origen y el destino. Los pesos se asignarán de acuerdo a la Figura 1, que muestra la relación empírica entre la tasa de error de trama (*Frame Error Rate*, FER) y la SNR recibida; dicha relación se basa en una extensa campaña de medidas [12], con más de 150000 tramas, con un tamaño de 1500 octetos, para reflejar el peor caso posible. Como se puede ver se establecen tres 'estados' diferentes, asignando el valor 1 a los canales *'buenos'*, 3 a los enlaces *'medios'*, mientras que el 7 se le aplica a los canales *'malos'*, de tal manera que, por ejemplo, una ruta de dos enlaces *'buenos'* se favoreciera sobre un único salto de calidad *'media'*. Es importante destacar el hecho de que esta asignación se hace únicamente en función de la SNR recibida, que es monitorizada por la mayoría de interfaces inalámbricas comerciales. Además, gracias a utilizar un número reducido de estados, se evita la complejidad que caracteriza la métrica ETX, que define un número sensiblemente mayor de estados, aunque se sigue reflejando, de manera fidedigna el estado de los enlaces inalámbricos.

Seleccionando 14 y 8 *dB* como los dos umbrales que separan los estados, se asegura que la FER será menor del 10% para los canales *'buenos'* y que estará en el rango $[10, 50]\%$ para aquellos considerados de calidad *'media'*. Para favorecer la monitorización de la calidad de los diferentes enlaces, los nodos participan en un esquema de difusión de mensajes de *Hello*, como se propone, por ejemplo en [3];

además hay que tener en cuenta que muy posiblemente la mayoría de las redes inalámbricas multi-salto se beneficien de un protocolo genérico de monitorización de vecinos, conocido como *MANET Neighborhood Discovery Protocol* (NHDP) [13].

Se promedian las últimas ω muestras de la SNR, para estimar la calidad del enlace, llevando a cabo un promediado ponderado móvil. Además, para evitar fluctuaciones no deseadas en el estado de un enlace, ya que derivarían en el envío innecesario de mensajes de control, se implementa un sencillo procedimiento de histéresis, que reduce significativamente la tasa de cambio de estados. A pesar de que Aguayo *et al.* [14] postulan que la SNR podría tener poco valor a la hora de calificar la calidad de un enlace, se basan en un conjunto de medidas llevadas a cabo sobre una plataforma exterior, cuyo comportamiento es sensiblemente diferente al que se observa en entornos interiores. De hecho, De Bruyne *et al.* [15] (y las referencias en dicho trabajo) defienden la utilidad de la SNR como parámetro apropiado para estimar adecuadamente la calidad de enlaces inalámbricos en entornos interiores.

Además de la calidad individual por enlace ($\phi \in \{1, 3, 7\}$), también es necesario introducir un nuevo parámetro, referido a la calidad total de una ruta, que se define como $\mathbf{\Phi} = \sum_{i=1}^{N} \phi_i$, donde N es el número de saltos de la ruta completa. Finalmente, también se introduce la calidad media de una ruta ($\bar{\mathbf{\Phi}}$), definido como el cociente entre la calidad total de la misma ($\mathbf{\Phi}$) y el número de saltos (N).

B. Procedimiento de descubrimiento de ruta

El procedimiento de descubrimiento de ruta, que es inicializado por un nodo cuando quiere encontrar un camino a un destino, y no conoce ninguna alternativa válida, es muy similar al especificado por el protocolo DSR original. La diferencia más importante es que en el caso del SADSR la calidad del enlace con el salto anterior se incorporará al paquete *Route Request* (RREQ) a medida que viaja por la red, de acuerdo a la información que un nodo dispone cuando reenvía dicho paquete. Otra modificación relevante que se tiene que llevar a cabo afecta al modo en el que un nodo decide o no propagar un RREQ; en el DSR original, los paquetes de búsqueda de ruta que ya hubieran sido procesados se descartan, mientras que en el SADSR, si la calidad global de la ruta que llega en el RREQ es mejor que la anteriormente almacenada para el mismo par *iniciador-identificador RREQ*, el paquete se volverá a reenviar. Este mecanismo, que asegura que la ruta con mejor calidad global ($\mathbf{\Phi}$) siempre llegue al destino, puede causar cierta sobrecarga, debido al reenvío masivo de RREQ por parte de los nodos intermedios. Para evitar este efecto, se aleatoriza el proceso de propagación de RREQ; así, basándose en la calidad media de la ruta ($\bar{\mathbf{\Phi}}$), un nodo seleccionará un tiempo de espera, uniformemente distribuido en el intervalo $\left[\frac{\overline{\Phi}\left(\overline{\Phi}-1\right)}{2}, \frac{\overline{\Phi}\left(\overline{\Phi}+1\right)}{2}\right]$. Durante este tiempo el paquete se mantiene en un buffer local antes de ser reenviado y, si recibiera un RREQ que hubiera viajado por una ruta de mejor calidad (mayor $\mathbf{\Phi}$), con un paquete esperando en dicho buffer, dicha retransmisión se cancelaría, programando una nueva de acuerdo al valor del nuevo RREQ. De esta manera se asegura que los paquetes de búsqueda de ruta con mejor calidad se propaguen antes y, además, se ha comprobado que este mecanismo reduce drásticamente el número de RREQ que se difunden por la red y que también consigue priorizar aquellas alternativas de mayor calidad.

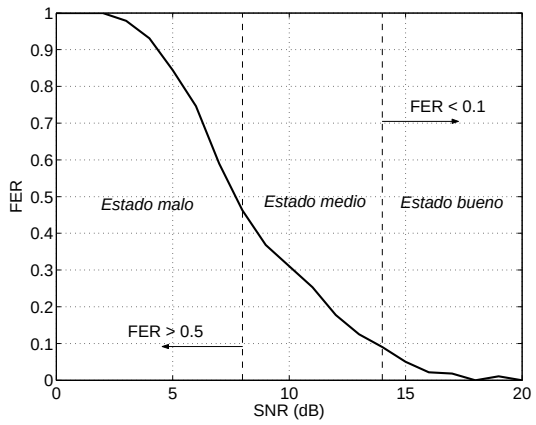


Fig. 1. Definición de estados para la métrica SADSR, en función de la SNR promediada por enlace

C. Procedimiento de mantenimiento de ruta

El SADSR también hace uso de la operación básica en el mecanismo de mantenimiento de ruta, ya que se asegura de comprobar la validez de los enlaces que se estén empleando, enviando mensajes de error de ruta (RERR) tras detectar la caída de algún enlace. Además, al emplear SADSR, un nodo también debe informar a la fuente correspondiente en una ruta activa (ruta por lo que está transmitiendo información) de un cambio en el estado de uno de los enlaces que la conforman. Así, es necesario mantener un registro de cuáles son las rutas activas en cada momento, de manera que un nodo pueda localizar las fuentes que deberían ser notificadas tras un cambio en la calidad de un enlace inalámbrico, que pudiera afectar a la calidad global de la ruta (ya sea mejorándola o empeorándola).

IV. Modelo de canal para interiores

Uno de los aspectos fundamentales en el análisis que se lleva a cabo en este trabajo es que se basa en un modelo de canal novedoso, capaz de reflejar de manera fidedigna el comportamiento que se observó empíricamente sobre canales inalámbricos reales, a partir de la SNR recibida, y de cómo ésta afecta a la probabilidad de que una trama se reciba o no con error. El *Bursty Error model based on Auto-Regressive filter* (BEAR) es un modelo de canal capaz de modular su comportamiento en función de la calidad de los enlaces, en términos de la SNR recibida por trama y, por tanto, se puede emplear para analizar técnicas de optimización cruzada entre capas, como la que se postula en este trabajo. BEAR es capaz de reflejar adecuadamente el comportamiento de entornos típicos de oficina, especialmente considerando la aparición de ráfagas de errores [12], [16], que aparecen con gran probabilidad en escenarios de propagación reales y que además pueden tener un efecto considerable sobre la estimación que lleva a cabo la métrica ETX, tal y como se recoge en [8].

El modelo BEAR emula la SNR a través de una combinación de varias contribuciones: la dependencia con la

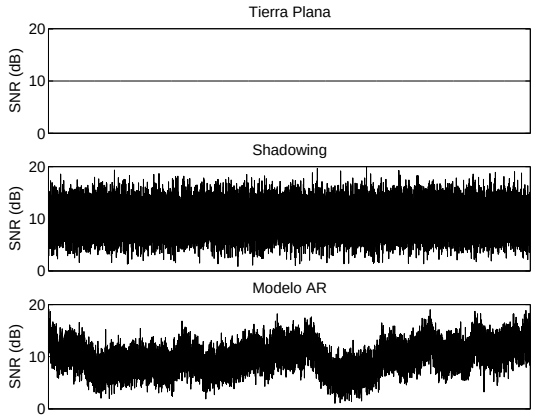


Fig. 2. SNR instantánea para varios modelos de canal

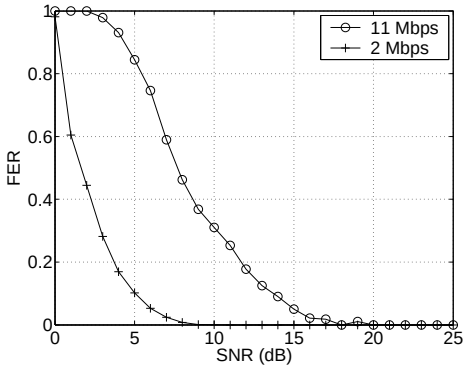


Fig. 3. Relación entre la FER y la SNR sobre un canal real para transmisiones *unicast* (@11 Mbps) y *broadcast* (@2 Mbps)

distancia entre transmisor y receptor, una variación lenta, que se modela a través de un filtrado AR y, finalmente, una variación rápida. Como se puede ver en la Figura 2, la SNR que proporciona el modelo BEAR es capaz de capturar cierta correlación entre los valores observados para tramas consecutivas, al contrario de lo que ocurre con enfoques más tradicionales utilizados habitualmente en entornos de simulación y que, bien son deterministas (*Tierra Plana*) o incorporan cierta componente aleatoria, aunque sin ninguna correlación entre tramas consecutivas (*Shadowing*). Además, es interesante destacar el hecho de que BEAR es capaz de configurarse de manera asimétrica, lo que aparece como una ventaja adicional, ya que algunas métricas (como ETX) asumen propagación simétrica.

Por otro lado, en BEAR se decide la existencia de error en una trama basándose en la SNR de la misma; así, y en lugar de utilizar el tradicional umbral fijo que usan diferentes plataformas de simulación, se propone emplear una función *Logística* a tramos, para derivar la probabilidad de que una trama sea errónea, tal y como se puede ver en la expresión que se muestra a continuación y que refleja, asimismo, el comportamiento observado en un entorno real. Los parámetros a, b, c, que son los característicos de este tipo de funciones, se pueden ajustar en función del comportamiento empírico observado, permitiendo diferenciar entre las transmisiones *unicast* y las *broadcast*.

$$\widetilde{\text{FER}} = \begin{cases} 1 & \text{SNR} < \text{lt}, \\ \dfrac{a}{1 + e^{b(\text{SNR}-c)}} & \text{SNR} \in [\text{lt}, \text{ht}], \\ 0 & \text{SNR} > \text{ht} \end{cases} \quad (1)$$

Además, tal y como se puede ver en la Figura 3, que muestra la relación entre la FER y la SNR (también basadas en medidas reales, en las que se procesaron un número relevante de tramas, > 100000), hay una diferencia clara en el comportamiento que se observa para las tramas *unicast*, que se transmiten al régimen binario máximo, y aquellas que son difundidas en modo *broadcast*, que lo hacen a 2 *Mbps*, como consecuencia del efecto *Gray Zones* [10]. A pesar de la gran diferencia que existe, es posible emplear la anterior ecuación, con una combinación de parámetros adecuada, para modelar el comportamiento de ambos tipos de transmisiones, dotando al modelo de una gran flexibilidad.

V. Comportamiento de las diferentes estrategias de encaminamiento

La comparativa entre los comportamientos obtenidos por las diferentes estrategias de encaminamiento analizadas, y las posibles mejoras que la propuesta SADSR puede aportar se llevarán a cabo sobre dos escenarios diferentes. El primero de ellos, que se basa en una configuración en cadena de los nodos, permitirá establecer una comparación detallada del comportamiento de los diferentes algoritmos, ya que es posible analizarlo en detalle; posteriormente este conocimiento se utilizará para extrapolarlo a situaciones más complejas. Para complementar el análisis, también se presentarán los resultados obtenidos sobre un entorno más realista, en el que un conjunto de nodos se despliegan de manera aleatoria en una superficie bidimensional. En ambos casos se analizará el comportamiento de un conjunto de algoritmos de encaminamiento, además del que se ha presentado en el artículo, que aparecen brevemente descritos a continuación.

- **DSR original.** Algoritmo DSR original, que busca minimizar el número de saltos necesarios para alcanzar el destino.
- **DSR con monitorización de vecinos.** Como se ha dicho anteriormente, es bastante probable que la mayoría de algoritmos de encaminamiento para redes MANET incorporen mecanismos para que un nodo pueda monitorizar su conjunto de vecinos. Es por tanto interesante analizar cuál es el efecto de estos procedimientos sobre la operación del DSR, especialmente teniendo en cuenta la influencia del efecto *Gray Zones*. Trabajos existentes en la literatura [3], [6] asumieron en sus análisis que el protocolo DSR tenía la capacidad de detectar vecinos.
- **ETX.** Se trata de la métrica para enriquecer los algoritmos de encaminamiento multi-salto que cuenta con mayor reconocimiento por parte de la comunidad científica, como ya se ha mencionado anteriormente. Sin embargo, no existen en la literatura análisis de la misma utilizando plataformas de simulación, por lo que resulta interesante incorporarlo al estudio.
- **ETX sin *Gray Zones*.** Ya se conoce que el efecto *Gray Zones* puede tener una influencia relevante sobre el comportamiento de la métrica ETX [11], ya que

Tabla I
PARÁMETROS DE LA SIMULACIÓN

Aspecto	Parámetro
Escenario A - Topología en cadena	
Separación $S : D$	$\{5, 10, 15, 20\}$ m
# de nodos	3
Escenario B - Bidimensional	
Área de simulación	60×60 m^2
# de nodos	$\{15, 20, 25, 30\}$
Parámetros comunes a ambos escenarios	
Modelo canal	*BEAR* con Gray Zones (indicado si no lo está)
MAC	IEEE 802.11b (11 Mbps, 2 Mbps broadcast)
Movimiento de nodos	Estáticos
Tamaño paquetes	1500 B
Tráfico UDP	Poisson (10 pkt/s)
Tráfico TCP	Transferencia FTP 10 MByte
# de simulaciones	100

los paquetes que un nodo emplea para monitorizar la calidad de los enlaces se difunden en modo *broadcast*, por lo que pueden llevar a estimaciones optimistas; por tanto, teniendo en cuenta la gran flexibilidad que ofrece el modelo de canal BEAR, se configurará para que el tratamiento de los paquetes de *Hello* sea idéntico al que se le da a las tramas que transportan información. Es importante recordar que los trabajos que han analizado el comportamiento de ETX [3], [6] no consideraron este efecto, debido a la configuración particular de las tarjetas inalámbricas empleada.

La Tabla I resume la configuración empleada durante la evaluación de ambos escenarios.

A. Encaminamiento oportunista sobre una topología en cadena

En esta caso se sitúan los dos nodos origen y destino, S y D respectivamente, separados por una distancia d. Además se cuenta con un tercer nodo, que se sitúa entre ambos (aproximadamente en el punto medio entre ambos, con un cierto error de localización aleatorio). En este escenario, el nodo S transmite tráfico UDP, utilizando un modelo de *Poisson*, a una tasa de 10 paquetes por segundo. Cada medida se realiza en 100 ocasiones independientes, para obtener intervalos de confianza lo suficientemente pequeños.

Como se puede ver en la Figura 4, el algoritmo SADSR tiende a seleccionar rutas con un mayor número de saltos que el resto de alternativas. La idea principal que persigue esta propuesta es la de favorecer la utilización de enlaces con una calidad mayor (en términos de la SNR recibida), por lo que hay un elevado número de situaciones, especialmente al aumentar la distancia entre S y D, en los que el SADSR hace uso del nodo intermedio para reenviar el tráfico entre los extremos de la comunicación. Sin embargo, también se puede observar que cuando la condición del enlace directo es buena (cuando la distancia es de 5 metros, que se corresponde con un canal *'bueno'*), SADSR hace uso del mismo, y la presencia del tercer nodo es prácticamente testimonial. Además, se puede ver que únicamente el protocolo DSR original, sin la capacidad de detección de vecinos activada, hace un uso relevante del nodo intermedio (al menos cuando la separación entre el transmisor y el receptor es elevada). Para el resto de las estrategias analizadas, teniendo en cuenta el mayor alcance de los mensajes de *Hello*, la fuente siempre es consciente

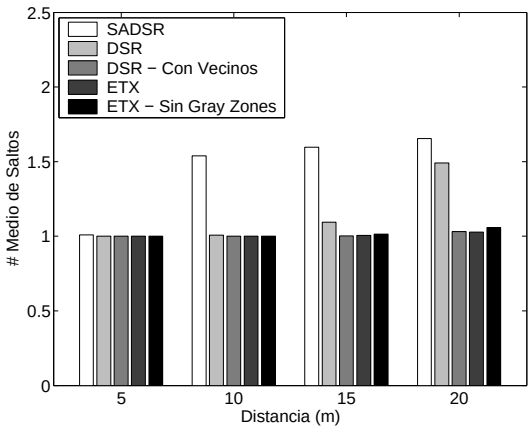


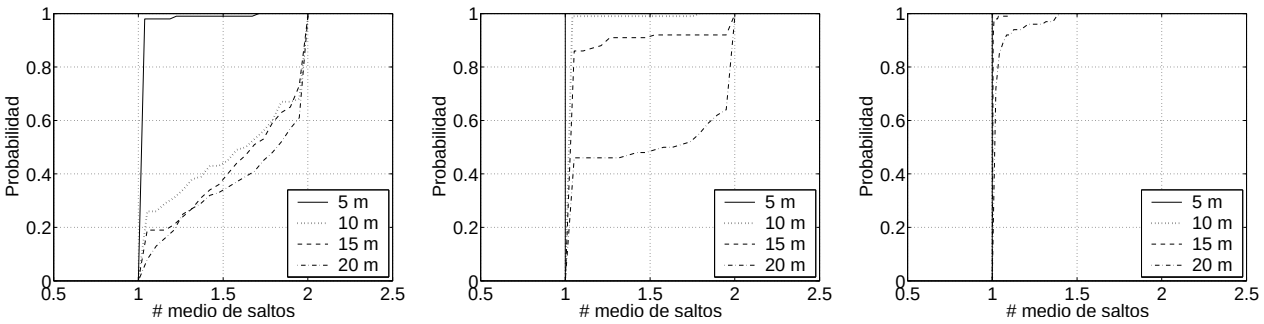
Fig. 4. Número medio de saltos en la topología en cadena oportunista

de la existencia de un enlace directo y, por tanto, insiste en hacer uso de la ruta de un único salto. Un aspecto a destacar es el relativo mal comportamiento de la métrica ETX, incluso al configurar el modelo de canal para evitar el efecto *Gray Zones*, ya que la estimación de la calidad de los enlaces debe ser realmente baja, para que derive en un valor ETX malo. La Figura 5 muestra las funciones de distribución de la longitud de la ruta media para cada una de las alternativas analizadas; como se puede ver, la métrica ETX (con las dos configuraciones del modelo de canal), así como el protocolo DSR con la capacidad de detección de vecinos activada, favorecen el uso de rutas de un único salto, mientras que tanto SADSR como la versión original de DSR hacen uso de la presencia *oportunista* del tercer nodo en el escenario. Se observa, sin embargo, que mientras SADSR presenta un comportamiento 'continuo', usando prácticamente todos los posibles valores entre 1 y 2 saltos, DSR selecciona rutas de manera más discreta, ya que la mayoría de ocasiones hace una selección, que mantiene durante toda la simulación; así, se refleja la capacidad de SADSR de adaptarse a las condiciones puntuales en la calidad de los enlaces, mientras que DSR simplemente reacciona ante la caída del enlace que se esté utilizando en cada momento.

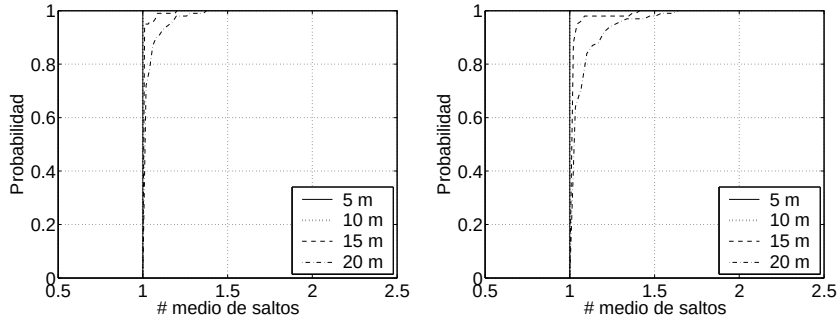
La Figura 6 muestra la manera en la que la política de selección de ruta se refleja en la tasa de error de paquete para las diferentes estrategias de encaminamiento analizadas. Como se puede ver, el protocolo SADSR claramente mejora el comportamiento del resto de alternativas, especialmente cuando la distancia entre los dos extremos de la comunicación es elevada. Por otro lado, sólo la versión original de DSR (sin la capacidad de detectar vecinos activada) es capaz de ofrecer resultados similares a los ofrecidos por SADSR, tal y como sucedía con las longitudes de las rutas empleadas.

B. Escenario bidimensional

Aunque el escenario anterior proporciona pruebas claras de las mejoras que la métrica SADSR puede aportar, es algo limitado, por lo que resulta interesante analizar si dicho comportamiento puede extrapolarse a situaciones más complejas y realistas. Para ello, se dispone de una superficie de 60×60 m^2, en la que se despliegan, aleatoriamente, un conjunto de nodos. Uno de ellos se sitúa siempre en el centro geográfico del escenario y, en cada una de las 100 simulaciones independi-



(a) Protocolo SADSR (b) Protocolo DSR, sin monitorización de vecinos (c) Protocolo DSR, con monitorización de vecinos



(d) Protocolo ETX, con efecto *Gray Zones* (e) Protocolo ETX, sin efecto *Gray Zones*

Fig. 5. Funciones de distribución de la longitud media de ruta para las diferentes estrategias de encaminamiento sobre la topología en cadena oportunista

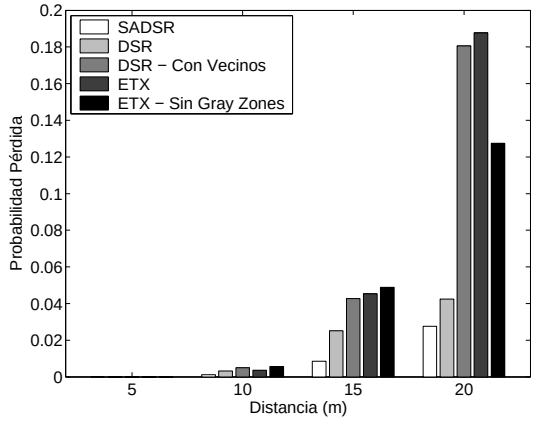


Fig. 6. Pérdida IP (PER) en la topología en cadena oportunista

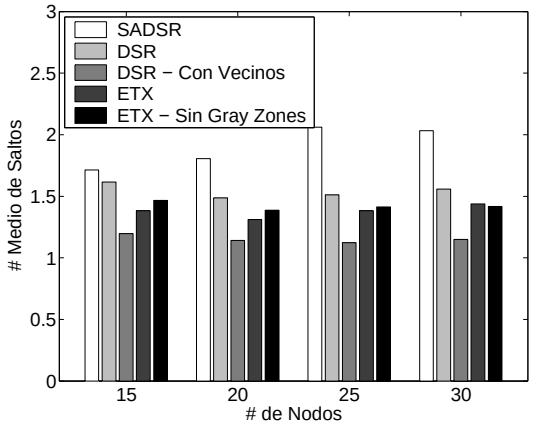


Fig. 7. Longitud media de ruta en el escenario bidimensional

entes, se elige un origen aleatorio para comunicarse con aquel. Bajo estas consideraciones, la Figura 7 muestra el número medio de saltos empleados en cada caso; como se puede ver, y como cabría esperar, el protocolo SADSR utiliza rutas más largas, haciéndose la diferencia más relevante cuando la densidad de los nodos se incrementa, ya que habría más posibilidades a la hora de establecer la comunicación y la métrica propuesta siempre trata de potenciar aquellas rutas con saltos más cortos (mayor calidad). Otra observación a destacar es que la longitud de la ruta parece crecer a medida que se incrementa el número de nodos desplegados, hasta que se estabiliza (no hay prácticamente diferencias entre los casos de 25 y 30 nodos); esta tendencia no se observa, sin embargo para el resto de alternativas analizadas. Por otro lado, la estrategia de encaminamiento que hace uso de rutas más cortas es el protocolo DSR original cuando tiene la capacidad de detectar vecinos activada, como consecuencia evidente del efecto *Gray Zones*.

La Figura 8 pone de manifiesto cómo la política de selección de ruta empleada se refleja en la pérdida de datagramas IP obtenida con cada uno de los protocolos. Como puede verse, la métrica SADSR presenta un comportamiento claramente superior al del resto de algoritmos y, además, es capaz de reducir la tasa de error de paquetes (PER) cuando el número potencial de rutas a emplear crece, esto es, cuando se despliegan más nodos en el escenario. Este es un hecho que no se pone de manifiesto para el resto de protocolos, ya que la PER permanece bastante estable para todas las densidades que se han analizado. Posiblemente, la conclusión más relevante es que el efecto *Gray Zones* tiene una consecuencia muy adversa

sobre el rendimiento del resto de alternativas, ya que tanto la métrica ETX (cuando BEAR sí que se configura para simular dicho efecto), así como el protocolo DSR con la capacidad de detectar vecinos, muestran un comportamiento claramente peor que el del resto de alternativas. Hay que tener en cuenta que la mayoría de los estudios experimentales que se han hecho acerca del comportamiento de la métrica ETX y su comparación con el protocolo DSR [3], [6] no tienen en cuenta el mayor alcance de las tramas *broadcast*, teniendo en cuenta la configuración particular de las tarjetas inalámbricas. En estas circunstancias, los resultados del simulador son adecuados, ya que el comportamiento del protocolo DSR con la capacidad de detectar vecinos activada es peor que el obtenido con la métrica ETX (configurando el modelo de canal BEAR para no tener en cuenta el efecto de las *Gray Zones*).

Finalmente, la Figura 9 compara el rendimiento TCP que se obtuvo con los diferentes protocolos de encaminamiento. De nuevo, SADSR aparece como la alternativa que ofrece un mejor comportamiento. Por otro lado, es interesante destacar que el protocolo DSR original (cuando no tiene la capacidad de vecinos activada), es capaz de ofrecer un comportamiento incluso mejor que el de la métrica ETX. De todas maneras, para reflejar más adecuadamente los análisis que se han llevado a cabo anteriormente, la comparativa tiene que realizarse entre el protocolo DSR cuando sí que puede detectar vecinos y la métrica ETX cuando el efecto *Gray Zones* no está presente, observándose que, bajo estas premisas, los resultados son comparables con los presentados en [3], [6].

VI. Conclusiones

En este trabajo se ha presentado una métrica novedosa para mejorar el encaminamiento sobre topologías multi-salto, basada en la SNR de los enlaces inalámbricos. Se ha demostrado que el protocolo SADSR es capaz de mejorar las prestaciones, no sólo de la versión original del DSR, sino que también presenta un comportamiento superior al de la métrica ETX, que ha suscitado un gran interés por parte de la comunidad científica.

La evaluación se ha llevado a cabo a través de un extenso proceso de simulación, en el que se ha empleado un modelo de canal realista, capaz de reflejar, de manera fidedigna, el comportamiento real observado en entornos típicos de oficinas. En este sentido, el modelo BEAR asegura la validez del análisis, ya que es capaz de emular los resultados que se hubieran obtenido en escenarios reales, además de ofrecer las ventajas tradicionalmente atribuidas a los estudios por simulación. Así, el trabajo analiza diversos parámetros, lo que permite establecer una comparativa adecuada del comportamiento de las diferentes estrategias analizadas. Los resultados asimismo ponen de manifiesto la influencia negativa que el efecto denominado como *Gray Zones* puede tener sobre el rendimiento de las diferentes estrategias de encaminamiento, mientras que no degrada la métrica propuesta en este trabajo.

El análisis se puede extender, incorporando cierta movilidad a los nodos, aprovechando la flexibilidad que un análisis mediante técnicas de simulación ofrece. Además también pueden añadirse nuevas métricas al proceso de selección de ruta (incluyendo la posibilidad de modificar el canal de transmisión de manera dinámica), para materializar el paradigma del encaminamiento *cognitivo*.

Agradecimientos

Los autores desean expresar su agradecimiento al Proyecto Nacional de I+D del Ministerio de Educación y Ciencia titulado '*Optimización de Técnicas de Descubrimiento de Servicios sobre Plataformas Inalámbricas Heterogéneas*' (TEC2006-05819)

Referencias

[1] D. B. Johnson, Y. C. Hu y D. A. Maltz. The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4. Request For Comments RFC4728, IETF, February 2007.

[2] D. de Couto, D. Aguayo, B. Chambers y R. Morris. Performance of multihop wireless networks: Shortest path is not enough. En *Proceedings of the First Workshop on Hot Topics in Networking*. October 2002.

[3] D. de Couto, D. Aguayo, J. Bicket y R. Morris. A high-throughput path metric for multi-hop wireless routing. En *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, páginas 134–146. ACM Press, 2003. ISBN 1-58113-753-2.

[4] A. Adya, P. Bahl, J. Padhye, A. Wolman y L. Zhou. A multi-radio unification protocol for IEEE 802.11 wireless networks. En *BroadNets*. 2004.

[5] R. Dube, C. D. Rais, K.-Y. Wan y S. K. Tripathi. Signal Stability-Based Adaptive Routing (SSA). *IEEE Personal Communications*, páginas 36–45, February 1997.

[6] R. Draves, J. Padhye y B. Zill. A comparison of routing metrics for static multi-hop wireless networks. En *SIGCOMM '04: Proceedings of the 9th annual international conference on Mobile computing and networking*. ACM Press, 2004. ISBN 1-58113-753-2.

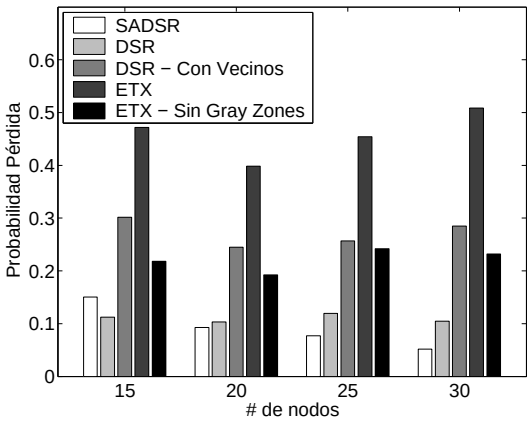


Fig. 8. Pérdida IP (PER) en el escenario bidimensional

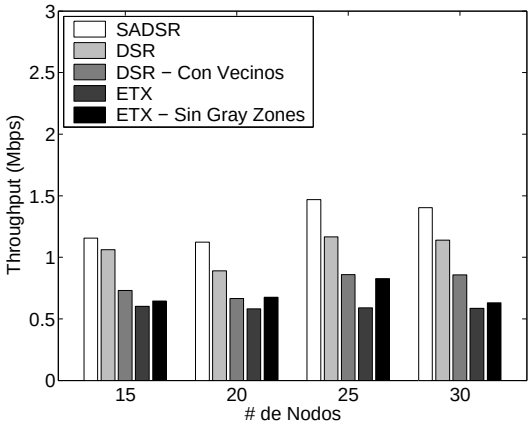


Fig. 9. Rendimiento TCP en el escenario bidimensional

[7] R. Draves, J. Padhye y B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. En *Mobicom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM Press, 2004. ISBN 1-58113-868-7.

[8] C. E. Koksal y H. Balakrishnan. Quality-aware routing metrics for time-varying wireless mesh networks. *IEEE Journal on Selected Areas on Communications*, 24:1984–1994, November 2006.

[9] B. Awerbuch, D. Holmer y H. Rubens. The Medium Time Metric: high throughput route selection in multi-rate ad hoc wireless networks. *Mobile Networks and Applications*, 11:253–266, 2006.

[10] H. Lundgren, E. Nordström y C. Tschudin. Coping with communication gray zones in IEEE 802.11b based ad hoc networks. En *5th ACM international workshop on Wireless mobile multimedia (WoWMoM 2002)*, páginas 49–55. ACM Press, 2002. ISBN 1-58113-474-6.

[11] K.-H. Kim y K. G. Shin. On accurate measurement of link quality in multi-hop wireless mesh networks. En *Mobicom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*. ACM Press, 2006. ISBN 1-59593-286-0.

[12] R. Agüero, M. García y L. Muñoz. BEAR: A bursty error auto-regressive model for indoor wireless environments. En *PIMRC 2007: The 18th Annual IEEE International Symposium on Personal, Indoor, and Mobile Communications*. Athens, Greece, 2007.

[13] T. Clausen, C. Dearlove y J. Dean. MANET neighborhood discovery protocol (NHDP). Internet Draft Version 4, IETF, June 2007.

[14] D. Aguayo, J. Bicket, S. Biswas, G. Judd y R. Morris. Link-level measurements from an 802.11b mesh network. *SIGCOMM Comput. Commun. Rev.*, 34(4):121–132, 2004.

[15] J. De Bruyne, W. Joseph, L. Verloock y L. Martens. Evaluation of link performance of an indoor 802.11g network. páginas 425–429. Jan. 2008.

[16] R. Agüero, M. García y L. Muñoz. On the accurate simulation of TCP behavior over error-prone wireless links with memory. En *WCNC 2008: IEEE Wireless Communications & Networking Conference*. Las Vegas, USA, 2008.

Utilización de Información de Nivel de Enlace para el Encaminamiento en Redes ad hoc

A. Ariza, A. Triviño, E. Casilari
Department of Electronic Technology (DTE)
University of Málaga
E.T.S.I Telecomunicación, Campus de Teatinos, Málaga,SPAIN
Email: aarizaq,atc,ecasilari@uma.es

J. C. Cano, C. T. Calafate, P. Manzoni
Department of Computer Engineering (DISCA)
Polytechnic University of Valencia
Camino de Vera, s/n, Valencia, SPAIN
Email: jucano,calafate,pmanzoni@disca.upv.es

***Abstract*—Mobile Ad Hoc Networks have a highly dynamic topology due to the terminal mobility. Ad hoc routing protocols can cope with this mobility as they search for an alternative route when a currently-in-use path breaks. Despite of their ability to recover from path failures, the time elapsed until the route is reestablished deteriorates network performance. MANET routing protocols can detect link breakages faster when link layer feedback (LLF) mechanisms are used. In this paper we evaluate the advantages and drawbacks of using feedback mechanisms. Our simulation results show that these mechanisms are appropriate for low mobility conditions. In contrast, feedback is not adequate for scenarios where terminals move faster due to frequent collisions that are mistaken as link breakages.**

I. Introducción

Una de las tecnologías que mayor interés está acaparando en los últimos años son las redes inalámbricas multisalto *Ad-Hoc*, también conocidas como redes MANET (*Mobile Ad-hoc Networks*). Estas redes usan una interfaz radio para el establecimiento de los enlaces con los nodos vecinos y son capaces de configurarse para formar redes complejas de una forma completamente autónoma. Para ello, se adaptan a los cambios de topología derivados tanto de las características de los enlaces radio como de la movilidad de los nodos, circunstancias que, frecuentemente, determinan las ruptura y creación de los enlaces. La autonomía de estas redes las hace especialmente adecuadas para diversas aplicaciones tanto militares como civiles.

Un aspecto básico para el funcionamiento de este tipo de redes lo constituyen los protocolos de encaminamiento. Estos protocolos son los encargados de buscar los caminos que permiten la comunicación entre cualquier par de nodos pudiendo usar nodos intermedios como *routers* para poder acceder a nodos que en principio están mas allá del alcance de la interfaz radio. Para conseguir este objetivo los protocolos de encaminamiento diseñados para este tipo de redes deben cumplir una serie de requisitos como son la capacidad de descubrir los nodos vecinos o la facilidad para adaptarse a los cambios en la topología de la red. Estas tareas deben implementarse manteniendo una baja tasa de generación de paquetes ya que el medio radio es un medio limitado e interesa que la señalización consuma pocos recursos a fin de dejar disponible al tráfico de datos el mayor ancho de banda posible.

Teniendo en cuenta estos dos objetivos, se han propuesto para estas redes un nuevo tipo de protocolos de encaminamiento: los protocolos reactivos. En ellos, la búsqueda del camino se realiza bajo demanda, esto es, en el momento en el que una fuente debe comunicarse con un destino y desconoce una ruta hacia él. En contraposición a este tipo de protocolos se encuentran los protocolos proactivos. En los protocolos proactivos los nodos de la red tienen en todo momento las tablas de todos los caminos posibles hacia cualquier destino dentro de la red. Esta información se mantiene con independencia de si estos caminos están en uso o no. En la literatura sobre redes MANET se han propuesto diversas implementaciones de protocolos de encaminamiento *ad hoc* [1] [2] [3] [4] [5]. De todas ellas, en la actualidad IETF (*Internet Engineering Task Force*) ha seleccionado tres protocolos reactivos y un protocolo proactivo. Entre los protocolos reactivos destacan por su rendimiento en redes con alta movilidad el protocolo AODV (*Ad Hoc On Demand Distance Vector*) [1] y su sucesor DYMO (*Dynamic MANET On-demand*) [4]. Dentro de los proactivos, OLSR (*Optimized Link State Routing*) [3] es uno de los más extendidos en la actualidad.

En este trabajo se pretende estudiar si las prestaciones de los protocolos de encaminamiento pueden mejorar al usar información estrictamente reservada al nivel MAC (*Medium Access Control*). En concreto, los mecanismos en los que se va a centrar este estudio son la promiscuidad radio y la detección de pérdida de conectividad del nivel MAC.

El objetivo de usar la promiscuidad se basa en el hecho de que el gasto energético de un nodo para recibir un paquete se realiza independientemente de si dicho paquete tiene como destino otro nodo. Por lo tanto, puede resultarle conveniente al nodo procesar cierta información del paquete aunque él no sea el destino. En concreto, al implementar la promiscuidad en los nodos de la MANET, los terminales serán capaces de aprender rutas adicionales.

Por otro lado, la detección de la pérdida de conectividad a nivel MAC, lo que se conoce como LLF (*Link Layer Feedback*), está basado en el hecho de que los paquetes *unicast* enviados por la subcapa MAC tienen confirmación de recepción mediante una trama ACK (*Acknoledgment*). En caso de no recibir esta confirmación, el nivel MAC reenviará el paquete un número predeterminado de veces. Si después de los diversos reintentos no se ha recibido la confirmación, el nivel MAC asume que se ha producido una pérdida de conectividad tras lo cual notificará a la capa superior que no ha podido confirmar la trama. Con esta información, a través de las tablas de encaminamiento, el protocolo de encaminamiento podrá determinar con qué nodo se ha perdido la conectividad. Como alternativa a este mecanismo, se encuentra el uso de mensajes *'Hello'*. Estos mensajes periódicos se envían

exclusivamente entre nodos vecinos. La periodicidad de este tipo de mensajes permite detectar cambios en los enlaces con los nodos vecinos. En este trabajo se evalúan estas técnicas de control de topología.

El resto de este trabajo está organizado como sigue. La sección II presenta una revisión de los protocolos de encaminamiento que se han empleado en este trabajo así como sus distintas configuraciones. La sección III describe el uso de información de nivel MAC por parte de los protocolos de encaminamiento. La evaluación de esta utilización se muestra en la sección IV. Finalmente, en la sección V se presentan las conclusiones de este trabajo.

II. Protocolos de encaminamiento MANET

Como se ha comentado anteriormente, en este trabajo se han considerado 3 protocolos de encaminamiento con diversas variantes relacionadas con el uso de información del nivel MAC. A continuación se detallan las características más importantes de estos protocolos.

A. AODV

Este protocolo [1] crea las rutas bajo demanda. Cuando un nodo necesita una ruta hacia un destino genera un mensaje RREQ (*Route Request*) que se propaga en la red con un proceso controlado de inundación. Cuando un nodo recibe este mensaje, lo retransmite si no conoce una ruta hacia el destino. En otro caso, responde al origen con un mensaje RREP (*Route Reply*). En el caso de que ningún nodo intermedio posea una entrada válida en la tabla de encaminamiento hacia ese destino, será el destino el que responda con un mensaje RREP cuando esté dentro de la red.

B. DYMO

La funcionalidad de DYMO[4] es parecida a la de AODV. La principal diferencia se centra en que en DYMO un nodo puede mantener múltiples rutas hacia un mismo destino. Para ello, tanto los mensajes RREQ como los RREP almacenan el conjunto de nodos por los que pasan. Gracias a esta información adicional, los nodos no sólo crean una ruta hacia el destino deseado sino que pueden aprender caminos adicionales hacia terminales intermedios. Con este comportamiento, se reduce la carga en la red y el tiempo para descubrir caminos.

C. OLSR

OLSR[3] es un protocolo de estado de enlace que se construye sobre OSPF [6]. Incluye, además, varias características para reducir la cantidad de paquetes de control en la red. Todos los *routers* mantienen información actualizada de la topología de la red. Para ello, los nodos intercambian mensajes periódicamente. Con el propósito de controlar la inundación de estos mensajes periódicos, un nodo determina los denominados MPR (*MultiPoint Relay*) a través de los cuales puede conocer la localización de los nodos que se encuentran a más de un salto.

Con OLSR no se emplea ninguna realimentación de la capa MAC. Los cambios de la topología se detectan, pues, con el intercambio de mensajes *'Hello'* periódicos. La ausencia de este tipo de realimentación se debe principalmente al hecho de que OLSR es un protocolo proactivo. Esto exige que las tablas de encaminamiento están permanentemente sincronizadas para evitar la formación de bucles en los caminos. En una red dinámica, en la cual los caminos están continuamente creándose y destruyéndose, una realimentación por parte del nivel MAC obliga a una continua inundación de la red de paquetes de actualización para tratar de conseguir la sincronización entre las tablas de encaminamiento de los distintos nodos, pudiendo llegar a saturar la red sin ni siquiera generar tráfico de datos.

III. Promiscuidad en DYMO

En este trabajo se ha estudiado el efecto de la realimentación por parte del nivel de enlace en el rendimiento final de los protocolos AODV y DYMO. Así mismo se ha estudiado la posibilidad de implementar un mecanismo de promiscuidad para los mensajes RREP en el protocolo DYMO.

Mediante el mecanismo de promiscuidad implementado, los terminales de la red puedan analizar paquetes que no están destinados hacia ellos. Si estos paquetes contienen información sobre la ruta por la que han sido propagados, los nodos podrían aprender rutas adicionales. Esto es lo que ocurre cuando DYMO está configurado con promiscuidad. En concreto, los mensajes adicionales que van a analizar los nodos van a ser los RREP. Como los niveles de enlace de los nodos deben recibir y procesar los paquetes antes de poder descartarlos en el caso de no ser los destinatarios de los mismos, los nodos de la red pueden obtener información adicional de estado procesando todos los paquetes de encaminamiento sin un consumo adicional de energía. En la figura 1 se muestra un ejemplo de cómo funciona la promiscuidad. En este ejemplo el nodo S desea establecer comunicación con el nodo D, por lo que inicia un ciclo de descubrimiento de ruta enviando un mensaje de RREQ. El nodo D recibe el mensaje y responde con un RREP. Tanto el mensaje RREQ como el RREP llevan el conjunto de nodos por los que ha pasado el mensaje. Los mensajes RREQ son enviados a la dirección *broadcast* por los que son recibidos y procesados por todos los nodos que obtienen información útil de ellos. Por el contrario, los mensajes RREP son enviados en forma *unicast*, por lo que sólo el nodo al que va dirigido podría procesarlo. Al usar el modo promiscuo no sólo el nodo al que va dirigido el mensaje lo procesaría, sino también todos los nodos que reciben el mensaje. En el caso de la figura 1, cuando el nodo 2 envía el mensaje hacia el nodo S, los nodos 5 y 1 también recibirían y procesarían el mensaje y aprenderían las rutas contenidas en el mensaje. Por su parte los nodos 3 y 4, que también recibirían el mensaje lo descartarían, el nodo 4 por estar su dirección en el mensaje, y el nodo 3 al haber recibido el mensaje RREP cuando lo envió el nodo 4. El nodo 3, que recibió y proceso anteriormente el mensaje RREP (cuando fue enviado por el nodo 4) tiene en sus base de datos un camino más corto, por lo que no modificará su tabla de rutas. Este mecanismo de promiscuidad permite también descubrir adyacencias por lo que podríase procesar cualquier mensaje recibido por el nivel MAC de los nodos vecinos, sin necesidad de que este sea un mensaje del protocolo *DYMO*. De esta forma, con un simple paquete *RREP*, no dirigido a él, un nodo puede descubrir otras rutas adicionales.

Con estas dos opciones, las variantes a comparar de los protocolos son las que se muestran en la Tabla 1.

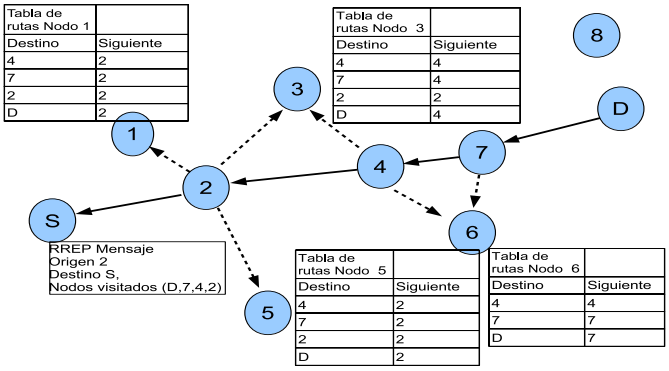


Fig. 1. Promiscuidad en DYMO.

AODV Hello
AODV Link layer Feedback
DYMO Hello
DYMO Link layer Feedback
DYMO Link layer Feedback con promiscuidad
OLSR

TABLA I
VARIANTES DE LOS PROTOCOLOS USADAS EN LAS PRUEBAS.

IV. PRUEBAS Y RESULTADOS

A. Escenario

El escenario usado en las pruebas consiste en un rectángulo de 1750x750 metros. Los nodos no se mueven libremente a lo largo del escenario sino que su movimiento está restringido a una serie de caminos pre establecidos. En la figura 2 se muestra el escenario con sus caminos, este escenario ha sido generado con la herramienta patAdHoc [7].

El escenario consta de 50 nodos que se pueden mover libremente por los caminos establecidos. De esos 50 nodos 12 son fuentes que generan tráfico a ritmo constante de 4 paquetes por segundo. Cada fuente elige como destino, para cada uno de los paquetes generados, uno de los 49 nodos restantes de forma equiprobable. Para el nivel *MAC* se ha usado la norma 802.11g fijando la velocidad de transmisión a 54 Mbits/s y una distancia máxima de transmisión 250 metros usando el modelo de espacio libre. Para el modelo de interferencias se ha empleado un modelo aditivo. La duración de cada simulación se ha limitado a 1000 segundos,

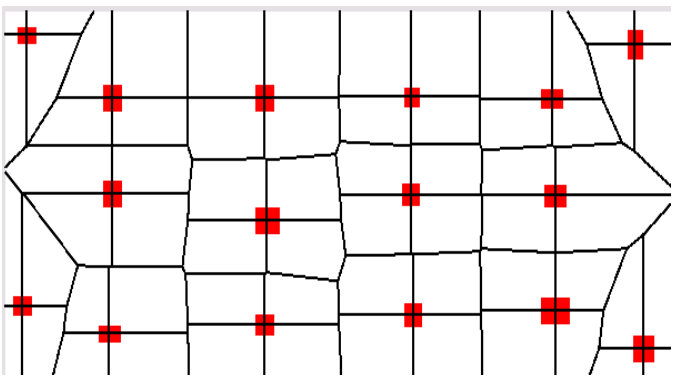


Fig. 2. Escenario de pruebas con los caminos válidos.

repitiendo 5 veces cada simulación en cada escenario con distinta secuencia aleatoria. Para el modelo de error de canal se ha usado las tablas generadas por Matteo Trivellato [8]. Se han realizado diversas pruebas modificando la velocidad (constante) de los nodos.

El simulador usado es el OMNET++ [9] con el *framework* de simulación inetmanet descargable en http://github.com/inetmanet/inetmanet/tree/master

B. Resultados

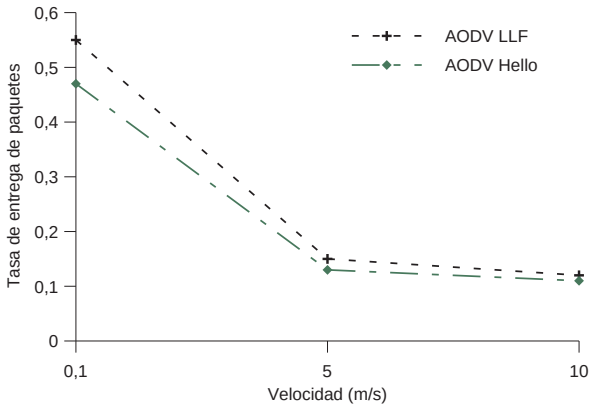
Los resultados analizados en esta sección son: retraso medio de paquetes, que es el tiempo medio que tardan los paquetes, que son recibidos por el destino, en llegar a este; tasa de entrega de paquetes, que es el porcentaje de paquetes recibidos correctamente por el nivel de aplicación del nodo destino, sobre el total de paquetes enviados; y la tasa de colisiones, que son los paquetes no han podido ser recibidos correctamente por el nivel de enlace debido a colisiones producidas por las transmisiones de otros nodos.

En la figura 3 se muestra la tasa de entrega de paquetes. Se aprecia que el protocolo OLSR presenta un excelente comportamiento en entornos de baja movilidad ya que la estrategia proactiva permite un conocimiento completo de la red. Por ello, presenta un excelente rendimiento al usar de una forma mas eficiente los recursos de la red.

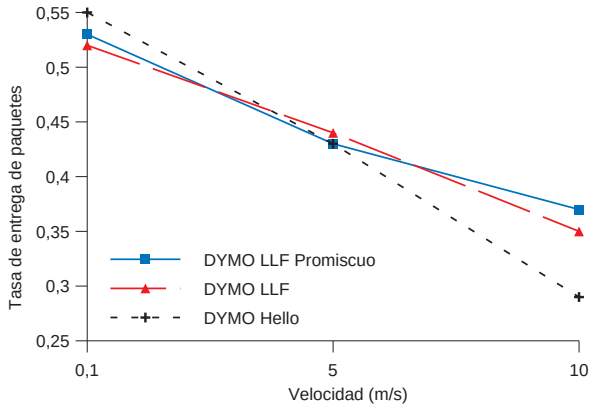
Por el contrario, la movilidad degrada rápidamente el funcionamiento del protocolo OLSR debido al tiempo que precisa para detectar los cambios en la conectividad de la red mediante el mensaje *Hello*, así como el tiempo que se precisa para propagar la información de estado a toda la red. Este fenómeno también es apreciable en el caso de los protocolos *AODV* y *DYMO* cuando usan el mismo mecanismo (figuras 3 y 4), precisa un elevado tiempo para detectar los cambios en la red. Así, si la movilidad de la red es elevada se tarda demasiado en detectar la pérdida de adyacencia, lo que se traduce en unas elevadas pérdidas y un considerable retraso. Se aprecia que los mecanismos LLF mejoran el rendimiento del protocolo en cualquier caso ya que permiten reconstruir la ruta mucho más rápidamente. Un fenómeno que beneficia al protocolo *DYMO* en escenarios con alta movilidad, (y al mismo tiempo lo perjudica en caso de baja movilidad), es que la respuestas a la petición de ruta siempre la realiza el destino.

En condiciones de alta movilidad la reconstrucción de la ruta desde un nodo intermedio da lugar a la rutas peores de las obtenidas cuando ésta se construye desde el nodo origen lo que provoca un mayor consumo de recursos al usarse rutas más largas, y observándose en este caso un mayor número de colisiones.

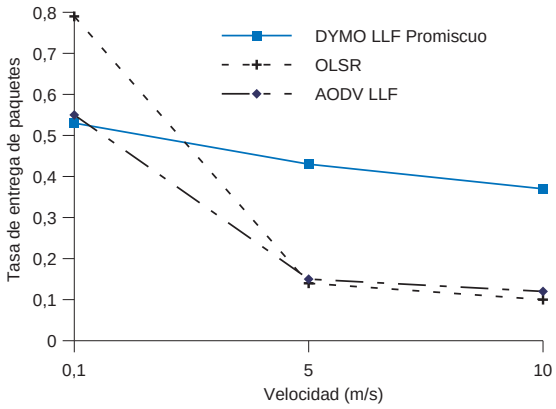
Un comportamiento a destacar cuando se analiza retraso de paquete es mal comportamiento del protocolo *AODV*, debido entre otras causas al mecanismo *local recovery*. Con este mecanismo, en el caso de que un nodo intermedio detecte una pérdida del enlace, los paquetes se almacenan mientras que se busca una nueva ruta. Este comportamiento da lugar a que algunos paquetes lleguen con un elevado retraso. Los otros protocolos no implementan este mecanismo, por lo que los nodos intermedios no almacenan paquetes mientras buscan nuevas rutas, directamente los descartan. De esta manera, disminuyen la varianza del retraso y el retraso medio. Se aprecia también, como era de esperar, que en situaciones de



(a) Protocolo *AODV*

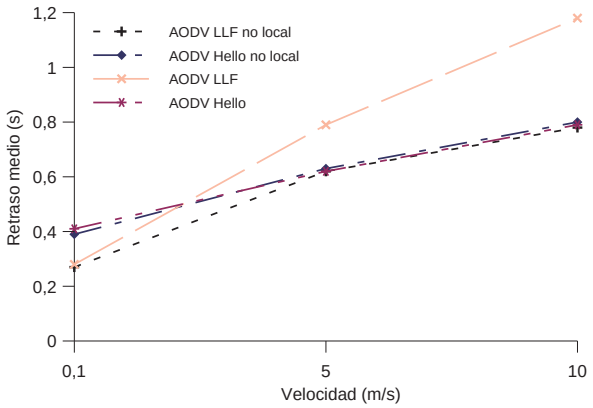


(b) Protocolo *DYMO*

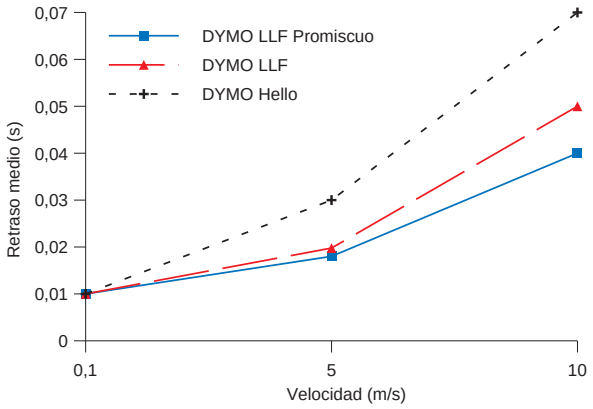


(c) Comparación *AODV*, *DYMO* y *OLSR*

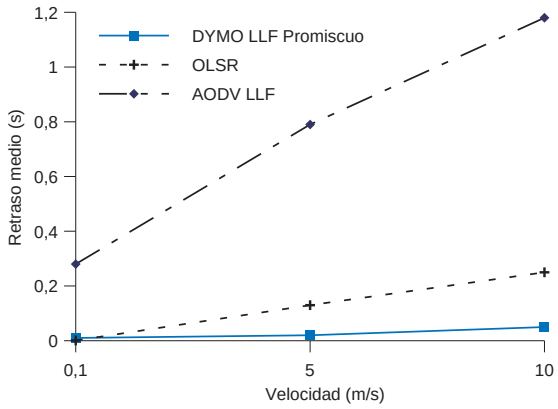
Fig. 3. Tasa de entrega de paquetes para las distintas versiones de *AODV* y *DYMO* y comparación entre protocolos



(a) Protocolo *AODV*



(b) Protocolo *DYMO*



(c) Comparación *AODV*, *DYMO* y *OLSR*

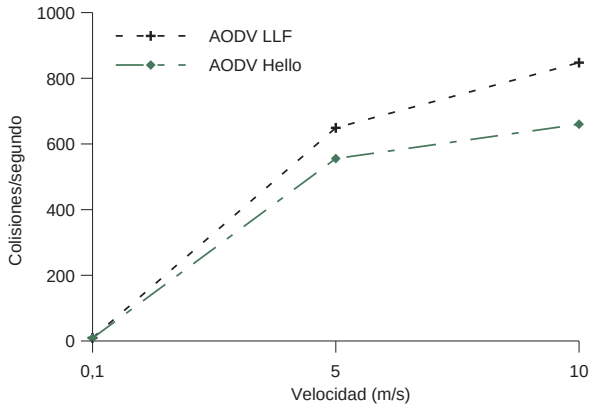
Fig. 4. Retraso medio

baja movilidad el protocolo *OLSR* tienen un bajo retraso, aumentando conforme aumenta la movilidad. También es posible apreciar, que en caso de alta movilidad, en el caso del protocolo *DYMO*, el retraso mejora con la aplicación de mecanismos *LLF* ya que permite reaccionar antes la pérdida de conectividad.

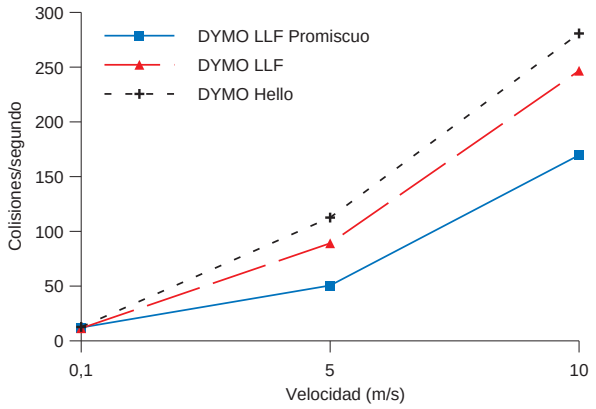
Finalmente, resulta también interesante analizar el comportamiento a nivel *MAC* y estudiar qué protocolo genera un menor número de paquetes a nivel *MAC* lo que se traduce en un mejor rendimiento energético. La figura 5 representa el número medio de colisiones por segundo, las colisiones suponen mensajes que no han podido recibirse correctamente y que han precisado múltiples reintentos. Un elevado número de colisiones implica una mayor sobrecarga del nivel de enlace y un mayor consumo energético. En este caso destaca el comportamiento del protocolo *DYMO* con *LLF* y promiscuidad. Este protocolo, que es el que mayor información recibe por parte del nivel 2 presenta un buen rendimiento a nivel de aplicación con movilidad, y al mismo tiempo, introduce la menor sobrecarga al nivel *MAC*.

V. Conclusiones

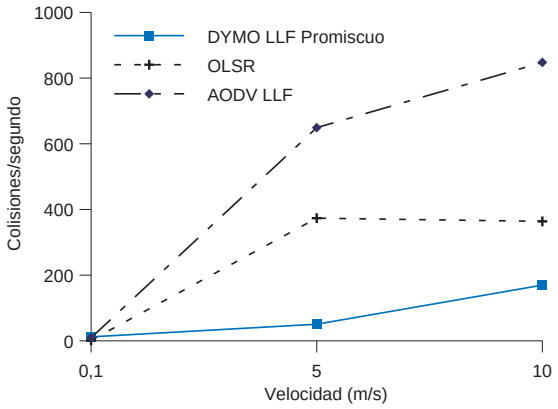
De los resultados se puede extraer una conclusión evidente: tener información por parte del nivel 2 mejora el rendimiento de los protocolos de encaminamiento en condiciones de



(a) Protocolo *AODV*



(b) Protocolo *DYMO*



(c) Comparación *AODV*, *DYMO* y *OLSR*

Fig. 5. Colisiones por segundo en la subcapa *MAC*

movilidad.

El menor rendimiento en condiciones de baja movilidad de las variantes con una mayor realimentación por parte del nivel 2 son debidas a los falsos positivos por las colisiones. Estos problemas tendrían solución si la realimentación por parte del nivel 2 fuera mayor. Una opción podría ser considerar que si la potencia recibida de un nodo se mantiene más o menos constante, las pérdidas serán por colisiones, no por pérdidas de conectividad, lo cual podría evitar que se procesen como rotura del enlace las pérdidas de paquete debidas a las colisiones. Además, si la información recibida por parte del nivel 2 fuera mayor e incluyera parámetros como la potencia recibida y la relación señal ruido, sería posible escoger caminos con una tasa menor de colisiones, mejorando el rendimiento final.

Es evidente que para mejorar el rendimiento de las redes *Ad-Hoc* los protocolos de encaminamiento necesitan una mayor realimentación por parte del nivel 2. La mejor forma para conseguir esto es trasladar el protocolo de encaminamiento del nivel 3 al nivel 2. Implementar el protocolo de encaminamiento a nivel 2 introduce una serie de ventajas.

- Mayor nivel de información para establecer los caminos. Es posible medir la potencia recibida y la relación señal ruido.
- Información “gratuita”. Cualquier paquete recibido, tenga como destino dicho nodo u otro puede ser usado para refrescar la conectividad. Esto evitaría la necesidad de usar mensajes tipo *Hello*.
- Extensión de la red. Extender el área de cobertura de la red sería mucho más sencillo. Bastaría con introducir repetidores de nivel 2 para ello. Estos repetidores serían completamente transparentes para los nodos con funcionalidad completa.
- Simplificación de la red. La complejidad de la red se reduce, los nodos *Ad-hoc* se verían como un conjunto de nodos a un salto, con conectividad completa, de modo similar al caso en que todos los nodos estuvieran conectados en un mismo segmento Ethernet.

Referencias

[1] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing,” RFC 3561 (Experimental), July 2003.
[2] R. Ogier, F. Templin, and M. Lewis, “Topology Dissemination Based on Reverse-Path Forwarding (TBRPF),” RFC 3684 (Experimental), Feb. 2004.
[3] T. Clausen and P. Jacquet, “Optimized Link State Routing Protocol (OLSR),” RFC 3626 (Experimental), Oct. 2003.
[4] I. Chakeres and C. Perkins, “Dynamic manet on-demand routing protocol (dymo), internet draft,” Published Online, 2008.
[5] D. Johnson, Y. Hu, and D. Maltz, “The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4,” RFC 4728 (Experimental), Feb. 2007.
[6] J. Moy, “OSPF Version 2,” RFC 2328 (Standard), Apr. 1998.
[7] E. Casilari A. Triviño, R. Morales-Berrocal, “Simulation of realistic mobility patterns for mobile ad hoc networks,” in *7th WSEAS Int.Conf. on APPLIED COMPUTER SCIENCE (ACS’07)*, Venice (Italy), 2007.
[8] University of Padova, “Ns-miracle: Multi-interface cross-layer extension library for the network simulator,” Available at http://www.dei.unipd.it/wdyn/?IDsezione=3966.
[9] A. Varga., “Omnet++ discrete event simulation system,” Available: http://www.omnetpp.org.

Selección de Enlaces Estables en redes MANET

Alicia Triviño Cabrera, Jorge García de la Nava, Eduardo Casilari
Departamento de Tecnología Electrónica,
Universidad de Málaga
Campus Universitario de Teatinos, Málaga
{atc, ecasilari}@uma.es

***Resumen*- Las redes móviles ad hoc se caracterizan por poseer una topología altamente dinámica debido a los frecuentes cambios en los estados de los enlaces inalámbricos. Los protocolos de encaminamiento son los encargados de descubrir y mantener las rutas de manera que cuando una ruta que se está empleando se rompe, se inicia el proceso para descubrir una ruta alternativa por la que continuar la comunicación. Los procedimientos de descubrimiento de ruta suelen provocar una inundación controlada de paquetes de control en la red, lo que ocasiona pérdidas y retardo en los paquetes de datos. Por lo tanto, las prestaciones de la red pueden mejorar considerablemente si se evitan estos procedimientos. Para ello, una estrategia a seguir es el uso de aquellos enlaces que van a durar más tiempo, esto es, los que son más estables. A partir de un modelo analítico, este artículo presenta un criterio para identificar los enlaces más duraderos. La evaluación mediante simulación muestra la habilidad del criterio para seleccionar los enlaces más estables.**

***Palabras Clave*-MANET, estabilidad, enlaces, duración de enlace.**

I. INTRODUCCIÓN

Las redes móviles ad hoc o MANET (*Mobile Ad Hoc Network*) ofrecen la posibilidad de que terminales distantes se comuniquen siempre y cuando exista una secuencia de terminales intermedios entre los extremos que puedan retransmitir los paquetes hasta el destino final. Los protocolos de encaminamiento ad hoc son los encargados de seleccionar los terminales intermedios así como de detectar la ruptura de los enlaces de una ruta. En los protocolos ad hoc reactivos, la ruptura de un enlace que está siendo utilizado exige la búsqueda alternativa de una ruta por donde pueda continuar el tráfico. El proceso de descubrimiento de ruta fuerza la emisión de paquetes de control que consumen los recursos energéticos de los nodos al mismo tiempo que ocupan parte del ancho de banda inalámbrico, ya de por sí escaso. Adicionalmente, durante el proceso de descubrimiento de la ruta alternativa, los extremos implicados no pueden comunicarse por lo que los paquetes de datos sufren retardos y, en algunas ocasiones, pérdidas. Con el propósito de evitar estos procesos, sería recomendable que los protocolos de encaminamiento ad hoc seleccionasen aquellos enlaces que van a durar más tiempo, esto es, los enlaces más estables.

A pesar de las ventajas potenciales que podrían obtenerse con la aplicación de este tipo de criterios, identificar el enlace que va a durar más tiempo no es una tarea trivial en la mayoría de las aplicaciones reales de las redes MANET donde los nodos se mueven de manera impredecible. Es por ello que el trabajo relacionado con este campo opta por abordar este problema con dos estrategias diferentes. Por un lado, algunos trabajos restringen la movilidad de los nodos de manera que se pueda obtener una caracterización analítica de la duración de los enlaces en redes MANET [1] [2] [3]. Debido a las restricciones impuestas, las conclusiones derivadas hay que utilizarlas con cautela en aplicaciones reales. Otra estrategia distinta para estimar la duración de los enlaces consiste en derivar algunas características estadísticas a partir de muestras de duraciones de enlace. Estas muestras pueden haberse obtenido de aplicaciones reales [4] o de simulaciones ejecutadas para este propósito [5]. En este trabajo, optamos por este tipo de aproximación para así poder analizar escenarios reales. Así pues, a partir de una caracterización estadística de la duración de los enlaces en MANET, se deriva analíticamente un criterio para identificar los enlaces que van a durar más tiempo.

El resto del artículo se estructura tal y como sigue. En la Sección II se describen otros criterios de selección de enlaces duraderos presentes en el trabajo relacionado. La Sección III muestra la derivación analítica del criterio que se propone en este artículo. La Sección IV explica los criterios que se evalúan en este trabajo para identificar los enlaces más estables. La Sección V muestra un escenario donde la aplicación del criterio resulta conveniente. Esta aplicación se evalúa mediante simulaciones tal y como se describe en la Sección VI. Por último, las principales conclusiones de este trabajo se resumen en la Sección VII.

II. TRABAJO RELACIONADO

Se distinguen dos tipos de criterios para seleccionar los enlaces más duraderos: los basados en potencia y los basados en la edad. Los criterios basados en potencia estiman el tiempo de vida restante de un enlace a partir de la potencia de la señal recibida a través de dicho enlace. Una de las primeras propuestas de esta categoría es SSA (*Signal Stability Adaptive*) [6]. En este protocolo los enlaces se clasifican en 'fuertes' o 'débiles' según la potencia de señal que se recibe por ellos. Cuando un nodo recibe un mensaje de RREQ (*Route Request*) como parte del proceso de descubrimiento de ruta, lo procesará sólo si lo recibe a través de un enlace 'fuerte'. De esta manera, las rutas que se descubren estarán compuestas exclusivamente por enlaces 'fuertes'.

Por otro lado, el trabajo en [7] estima la distancia que separa los dos extremos de un enlace por medio de la potencia de la señal recibida. Una vez que se estima la distancia, se infiere el movimiento de los nodos a través de la diferencia de

distancia en instantes consecutivos. El algoritmo define dos zonas concéntricas alrededor de cada nodo. Se asume que los nodos que se encuentran en la zona más cercana al centro (nodo analizado) mantienen enlaces estables con él. Como método de optimización de la red, por estos enlaces estables se enviará más tráfico. En contraposición, a los nodos que se encuentran en la zona exterior se les considera menos estables y se les retransmiten menos paquetes.

El principal inconveniente de los criterios basados en potencia es que dependen en gran medida de las condiciones de propagación que se asuman por lo que no son siempre aplicables en escenarios reales. Es por ello que se han propuesto los criterios basados en edad. En ellos se identifican los enlaces más duraderos a partir de su edad. En este grupo existen criterios contradictorios como la selección del enlace más joven [8], la selección del enlace más viejo [9] o la selección de un enlace de edad intermedia [10]. En este trabajo se analiza matemáticamente la estimación del tiempo de vida restante de un enlace a partir de su edad.

III. CRITERIO PROPUESTO PARA LA SECCIÓN DE ENLACES ESTABLES

La determinación del tiempo exacto de vida restante de un enlace es inabordable cuando los nodos se mueven de manera impredecible. Es por ello, que es necesario recurrir a heurísticos que proporcionen una aproximación del tiempo de vida restante de un enlace. En el campo de la fiabilidad, se usa el tiempo de vida residual medio o MRL (*Mean Residual Lifetime*) con el propósito de estimar el tiempo de vida restante de componentes electrónicos. Aplicado al contexto de estudio de este trabajo, el MRL de un enlace representa el tiempo de vida medio que se espera que un enlace siga activo dado que ya ha estado activo durante un periodo de tiempo o edad (*age*). Formalmente, el MRL de un enlace se define como [11]:

$$MRL(age)=E[LD-age/LD>age]=\frac{\int_{age}^{\infty}\overline{F}(u)du}{\overline{F}(age)} \quad (1)$$

donde *LD* representa la variable aleatoria 'duración de enlace' y $F(t)= 1 - \bar{F}(t)$ es la función de distribución de la variable *LD*.

En [12] se analizó que el mejor ajuste para la variable *LD* se correspondía con una función lognormal para escenarios caracterizados con distintos tipos de movilidad. Continuando con este estudio, en este trabajo modelamos la variable *LD* como una función lognormal de parámetros μ y σ por lo que el MRL de un enlace se calcula según la siguiente expresión:

$$MRL(age)=\frac{\int_{age}^{\infty}\left(\frac{1}{2}-\frac{1}{2}\mathrm{erf}\left(\frac{\log(u/\mu)}{\sigma\sqrt{2}}\right)du\right)}{\frac{1}{2}-\frac{1}{2}\mathrm{erf}\left(\frac{\log(age/\mu)}{\sigma\sqrt{2}}\right)} \quad (2)$$

Al resolver la integral, se obtiene que el MRL de un enlace equivale a:

$$MRL(age)=\frac{\frac{1}{2}\left[-e^{\frac{\sigma^2}{2}}\cdot\mu\cdot erf\left(\frac{\sigma^2-\log\left(\frac{u}{\mu}\right)}{\sqrt{2}\sigma}\right)+u\cdot erfc\left(\frac{log\left(\frac{u}{\mu}\right)}{\sqrt{2}\sigma}\right)\right]_{age}^{\infty}}{\frac{1}{2}-\frac{1}{2}\mathrm{erf}\left(\frac{\log(age/\mu)}{\sigma\sqrt{2}}\right)} \quad (3)$$

Como la expresión del numerador tiende a cero cuando el límite de la integral tiende a infinito, la ecuación anterior se puede simplificar en:

$$MRL(age)=\frac{\frac{1}{2}\left(e^{\frac{\sigma^2}{2}}\cdot\mu\cdot\left(1+erf\left(\frac{\sigma^2-\log\left(\frac{age}{\mu}\right)}{\sqrt{2}\sigma}\right)\right)-\mathrm{B}\right)}{\frac{1}{2}-\frac{1}{2}\mathrm{erf}\left(\frac{\log\left(\frac{age}{\mu}\right)}{\sigma\sqrt{2}}\right)} \quad (4)$$

$$B=age\cdot erfc\left(\frac{log\left(\frac{age}{\mu}\right)}{\sqrt{2}\sigma}\right) \quad (5)$$

Reduciendo el numerador y el denominador da lugar a la Ec. 6.

$$MRL(age)=\mu\cdot e^{\frac{\sigma^2}{2}}\cdot\left(\frac{1+erf\left(\frac{\sigma^2-\log\left(\frac{age}{\mu}\right)}{\sqrt{2}\sigma}\right)}{\mathrm{erfc}\left(\frac{\log(age/\mu)}{\sigma\sqrt{2}}\right)}\right)-age \quad (6)$$

Se ha comprobado la simplificación de la Ec. 6 comparando este resultado con la resolución de la Ec. 2 mediante métodos numéricos. Específicamente se ha empleado el método de cuadratura adaptativo de Simpson. La comparación demuestra que no existe diferencia entre ambas ecuaciones para distintos valores de μ y σ.

Una alternativa al uso de esta ecuación consiste en la construcción de un histograma. El histograma almacena los valores de la duración de enlace que un nodo percibe. A partir del histograma es fácil derivar los estimadores discretos de la función de densidad de probabilidad (*f[n]*) y de la función de distribución de probabilidad (*F[n]*) [9]. Con ellos, la función de distribución complementaria sería:

$$\bar{F}[n]=1-F[n] \quad (7)$$

Aplicando la siguiente ecuación, se puede calcular el MRL de un enlace a partir del histograma:

$$MRL[age]=\frac{\sum_{k=age}^{N}\bar{F}[k]}{\bar{F}[age]} \quad (8)$$

Este procedimiento, no obstante, presenta un inconveniente tal y como se muestra en la Figura 1 al comparar los resultados obtenidos de MRL con el histograma con los resultantes de la Ec. 6. Para la comparación, los valores μ y σ del ajuste lognormal de la función *LD* se obtienen a partir del cálculo de la media y la desviación estándar de las muestras de las duraciones de los enlaces

obtenidas en una simulación de 100000 segundos. El resto de los parámetros de la simulación se recoge en la Tabla 1.

Tabla 1. Parámetros de Simulación.

Área de simulación	1000 x 1000 m^2
Número de Nodos Móviles	50
Patrón de Movilidad	*Random WayPoint* Velocidad Mínima: 1 m/s Velocidad Máxima:[1-5] m/s Tiempo de Pausa: 0 s
Distribución inicial de nodos	Uniforme
Posición del Router de Acceso	(500, 500) m
Tiempo de simulación	100000s
Rango de transmisión	250 m
Modelo de propagación	Espacio Libre
Número de ejecuciones por punto	20

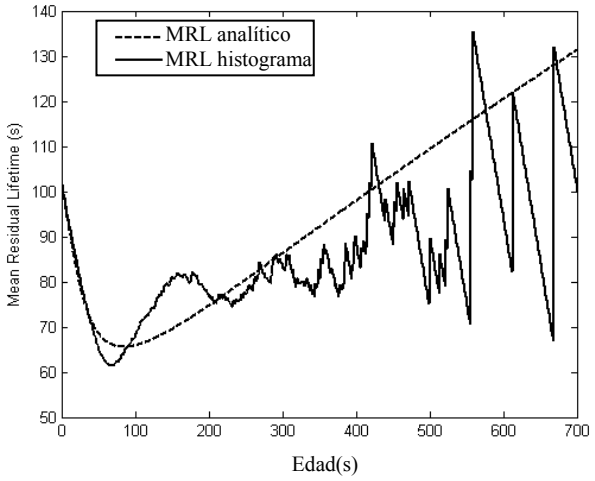


Fig. 1. Comparación de la resolución analítica y mediante histograma del MRL de los enlace para distintas edades.

En esta figura se aprecia que la utilización de un histograma provoca múltiples oscilaciones, especialmente cuando la edad de enlace se incrementa. Un comportamiento similar al representado en la Figura 1 se obtiene con distintas condiciones de movilidad lo que a equivale a distintos valores de los parámetros μ y σ.

IV. Criterios evaluados para identificar el enlace más estable

En este trabajo se ha evaluado la capacidad de cuatro estrategias para identificar el enlace más estable dentro de un conjunto de conexiones inalámbricas. La diferencia entre las estrategias reside en el método para calcular el tiempo de vida residual medio (MRL) de un enlace. Específicamente, se han empleado los siguientes cuatro algoritmos:

- Computación a partir de un histograma (MRL histograma). Para esta estrategia, los nodos construyen dinámicamente un histograma con las duraciones de enlace que observan. Además, los nodos mantienen la edad de los enlaces establecidos con sus vecinos.
- Computación asumiendo un ajuste exponencial de la variable *LD* (MRL Exponencial). Por su simplicidad, uno de los ajustes más utilizados para la duración de los enlaces en redes MANET se corresponde con una distribución exponencial [5]. Al ser un modelo sin memoria, el MRL de un enlace no depende de su edad sino que equivale al valor medio de la duración de los enlaces.
- Computación asumiendo un ajuste lognormal de la variable *LD* (MRL Lognormal). Este criterio se corresponde con el presentado en la Sección III donde el MRL de un enlace se calcula según la Ec. 6.
- Aleatorio. Con esta estrategia, todos los enlaces desde un nodo móvil al *gateway* comparten la misma probabilidad de ser seleccionados como el más estable independientemente de su edad o de la estimación del MRL.

V. Aplicación del criterio para la integración de una MANET con Internet

La integración a Internet de redes MANET exige el uso de un *gateway* [13] que complementa al *router* de acceso y está conectado a él. Entre las distintas implementaciones del *gateway*, algunas propuestas sugieren que para conseguir una integración de una red MANET en cualquier tipo de redes IP, las funcionalidades de *gateway* debería desempeñarlas un nodo de la MANET. De esta manera, la conexión de una MANET con redes externas no se limitaría exclusivamente a entornos previamente acondicionados con tal propósito, esto es, a entornos donde se haya instalado con antelación un *gateway*. Cuando un nodo de la MANET realiza las funcionalidades de *gateway*, recibe el nombre de *gateway* oportunista u ocasional. Para ejercer como tal, el nodo debe encontrarse en el área de cobertura del *router* de acceso para así poder encaminar en la MANET los paquetes procedentes de la red externa. Entre todos los nodos que se encuentran al alcance directo del *router* de acceso, uno se selecciona aleatoriamente para actuar como el *gateway*. El proceso para elegir un nuevo *gateway* se basa en un temporizador que cada nodo inicia con un valor aleatorio. Cuando el temporizador se agota, se observa si ya existe un nodo configurado como *gateway*. En caso contrario, el nodo se autoconfigura como el nuevo *gateway* y anuncia esta condición al resto de los nodos de la MANET.

El hecho de que un nodo de la MANET actúe como *gateway* no debería restringir su movilidad, tal y como se presenta en [14]. Bajo estas circunstancias, las funcionalidades de *gateway* se irán transfiriendo entre los nodos de la MANET de manera que cuando un *gateway* oportunista sale del área de cobertura del *router* de acceso, otro nodo de la MANET debe configurarse como el nuevo gateway oportunista. La transferencia de estas funcionalidades o *gateway switching* suele deteriorar las prestaciones de la red ya que durante el traspaso la red no dispone de ningún *gateway* y, por tanto, no puede comunicarse con el exterior [15]. Con el propósito de reducir los cambios de *gateways*, en este trabajo seleccionaremos como *gateway* al nodo que se encuentre en el área de cobertura del *router* de acceso y cuyo enlace con este

elemento se espera que vaya a durar más tiempo. Para esta selección, el valor de los temporizadores que usan los nodos para decidir si van a ser *gateways*, se inicializarán a un valor inversamente proporcional al tiempo que el nodo estima que va a durar el enlace que mantiene con el *router* de acceso. De esta manera, el nodo que comparta el enlace más estable con el *router*, se configurará como el nuevo *gateway* [14].

VI. Simulaciones y Resultados

La evaluación de la propuesta se ha realizado mediante simulaciones en MATLAB donde se controla la topología de la red con una matriz de adyacencia. Inicialmente, los nodos se distribuyen uniformemente en un área cuadrada de 1000x1000 m^2. El *router* de acceso se coloca en el centro de esta superficie y permanece estático. Por otro lado, los nodos se mueven siguiendo el modelo *Random WayPoint* [16]. Es uno de los modelos más utilizados en las simulaciones de redes MANET al tratarse de un modelo parsimonioso con el que la variación de pocos parámetros permite capturar múltiples condiciones de movilidad.

Respecto a la atenuación de la señal, el modelado se basa en la propagación en espacio libre por lo que dos nodos se encuentran directamente conectados siempre y cuando la distancia euclídea entre ambos no exceda el rango de transmisión. En este estudio, el rango de transmisión se ha fijado en 250 m.

Para tener en cuenta distintas condiciones de movilidad, la velocidad máxima de los nodos varía entre 1 y 5 m/s. Con el objetivo de obtener valores representativos, se han ejecutado 20 simulaciones para cada velocidad máxima analizada. Los resultados muestran el valor medio de las ejecuciones.

Las simulaciones se basan en el establecimiento de puntos de decisión. En cada punto de decisión, el programa selecciona un nodo para cada criterio evaluado dando lugar a cuatro nodos que podrían operar como *gateways*. Cuando el enlace entre el *router* de acceso y uno de estos cuatro posibles *gateways* se rompe, se calcula el tiempo en el que el nodo podría haber actuado como *gateway*. Este tiempo se almacena en una estructura de datos, denominado *TiempoActivación*, que es independiente para cada criterio. Una vez que se han roto los enlaces de los cuatro posibles *gateways*, se establece un nuevo punto de decisión y se repite el procedimiento.

Una vez finalizado el tiempo de simulación, las estructuras *TiempoActivación* permiten calcular el porcentaje de tiempo de simulación en el que los *gateways* de cada criterio han estado activos, esto es, conectados al *router* de acceso. De esta manera, se puede identificar qué criterio da lugar a *gateways* más duraderos. Esta medida se representa en la Fig. 2.

El resultado de la simulación refleja que la estimación del tiempo de vida residual basado en la derivación analítica presentada en este trabajo es capaz de detectar los enlaces más duraderos en todas las condiciones de movilidad estudiadas. Esta habilidad empleada en la selección de *gateways* oportunistas daría lugar a menos conmutaciones de *gateways* y, por tanto, las prestaciones de la red mejorarían.

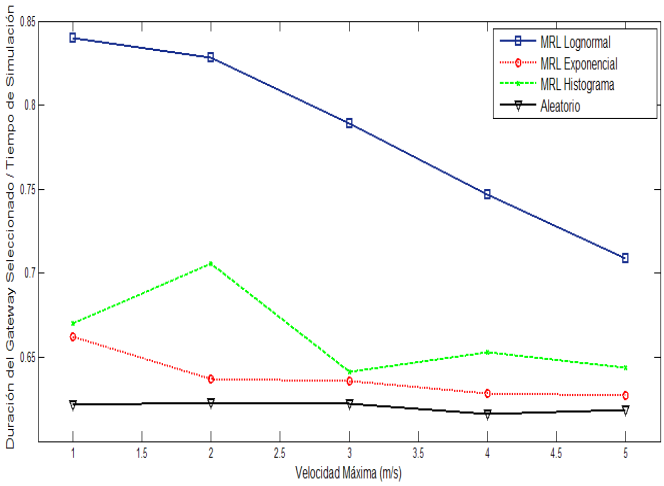


Fig. 2. Duración del *gateway* seleccionado para los cuatro criterios evaluados.

VII. Conclusiones

Los procedimientos de descubrimiento de ruta en una red MANET deteriora las prestaciones de la red por lo que es conveniente usar aquellos enlaces que van a durar más tiempo, esto es, los que son más estables. A partir del ajuste lognormal de la variable aleatoria de las duraciones de enlace en una MANET, este trabajo presenta la derivación analítica de un criterio que identifica el enlace más estable. La aplicación de este criterio en un MANET conectada con Internet muestra la bondad del mismo.

Referencias

[1] S. Cho, J. P. Hayes, "Impact of Mobility on Connection Stability in Ad Hoc Networks", IEEE WCNC, vol. 3, pp. 1650-1656, 2005

[2] X. Cheng, W. B. Heinzelman, "Exploring Long Lifetime Routing (LLR) in ad hoc networks", 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM), Octubre 2004.

[3] D. Turgut, S. K. Das, M. Chatterjee, "Longevity of routes in mobile ad hoc networks", IEEE Vehicular Technology Conference (VTC) Spring 2001.

[4] S. Bohacek, A. Ilic, V. Shidhara, "On the Predictability of Link Lifetimes in Urban MANETs", 4TH International Symposium on Modeling and Optimization in Mobile Ad Hoc and Wireless Networks (Wiopt), 2005.

[5] F. Bai, N. Sadagopan, B. Krishnamachari, A. Helmy, "Modeling path duration distributions in MANETS and their impact on reactive routing protocols", IEEE Journal on Selected Areas in Communications, vol. 22, Issue 7, pp. 1357-1373, Septiembre 2004.

[6] R. Dube, C. D. Rais, K. Y. Wang, S. K. Tipathi, "Signal Stability based Adaptive Routing for ad hoc mobile networks", IEEE Personal Communication, Febrero 1997.

[7] W. Zhu, M. Song, S. Olariu, "integrating Stability Estimation into Quality of Sevice Routing in Mobile Ad Hoc Networks", 14th IEEE Interntional Workshop on Quality of Service, pp. 122-129, Junio 2006.

[8] N. Meghanathan, "Comparison of Stable Path Selection Strategies for Mobile Ad Hoc Networks", International Conference on Networking, 2006.

[9] M. Gerharz, C. de Waal, P. Martini, P. James, "Strategies for finding stable paths in mobile wireless ad hoc networks", 28th Annual Conference on Local Computer Networks, pp. 130-139, Octubre 2003.

[10] E. Y. Hua, Z. J. Haas, "Path Selection Algorithms in Homogeneous Mobile Ad Hoc Networks", International Wireless Communications and Mobile Computing Conference (IWCMC), Julio 2006.

[11] N. Ebrahimi, "Estimation of Two Ordered Mean Residual Lifetime Functions", Biometrics, vol. 49, no. 2, pp. 409-417, Junio 1997.

[12] A. Triviño-Cabrera, J. García de la Nava, E. Casilari, F. J. González-Cañete, " Application of path duration study in multihop ad hoc networks", Telecommunication Systems, vol. 38, pp.3-9, Junio 2008.

[13] A. Triviño, "Integration of Mobile Ad Hoc Networks into IP-based Access Networks", Guide to Wireless Ad Hoc Networks, pp. 527-561, 2009.

[14] S.Singh, J.H. Kim, Y.G. Choi, K. L. Wang, Y.S. Roh, "Mobile Multigatway support for IPv6 mobile ad hoc networks", IETF Draft (trabajo en progreso), Junio 2004.

[15] A. Triviño-Cabrera, E. Casilari, F. J. González-Cañete, "Active gateway switching in hybrid ad hoc networks", Electronic Letters, vol. 42, no. 21, pp.1252-1254, Octubre 2006.

[16] J. Yoon, M. Liu, B. Noble, "Random Waypoint considered harmful", IEEE 22nd Annual Joint Conference of the IEEE Computer and Communications Society (INFOCOM), vol. 2, pp. 1312-1321, 2003.

Uso de un controlador difuso en redes MANET híbridas

A.J. Yuste

Departamento de Ingeniería de Telecomunicació
Universidad de Jaén
c/ Alfonso X El Sabio,28 23700 Linares (Jaén)
ajyuste@ujaen.es

A. Triviño, F.D. Trujillo, E. Casilari y A. Díaz-Estrella
Departamento de Tecnología Electrónica
Universidad de Málaga
Campus de Teatinos. 29071 MÁLAGA
{atc, fdtrujillo, ecasilari, adiaz}@uma.es

***Resumen*- Las redes móviles sin infraestructura o *Mobile Ad Hoc NETwork* (MANET) están formadas por nodos que se mueven unos de manera impredecible. Esta importante propiedad de las MANET implica que la topología de las mismas cambie frecuentemente. Los protocolos de encaminamiento ad hoc trabajan con esta topología dinámica pero para lograr conectar la MANET con redes externas como Internet es necesario un gateway.. El rendimiento de esta red híbrida (MANET e Internet) se ve afectado enormemente por la estabilidad, conectividad y la carga de la misma. La compleja interacción entre estos elementos puede ser modelada mediante conjuntos difusos. Este artículo presenta un sistema de control que utiliza técnicas de inteligencia artificial (algoritmos genéticos y sistemas difusos) para adaptar los mecanismos de localización o descubrimiento de los *gateways* por parte de los nodos móviles. El rendimiento del nuevo sistema difuso se evalúa mediante simulación y se compara con otros esquemas presentes en el trabajo relacionado.**

***Palabras Clave*- Internet, MANET , interconexión, algoritmos genéticos, sistemas difusos.**

I. INTRODUCCIÓN

El acceso completo y total de las redes inalámbricas es uno de los aspectos clave para lograr la computación móvil. En este sentido, ya existe un creciente interés en los sectores comerciales para favorecer el acceso a Internet en cualquier momento y en cualquier lugar. Sin embargo, este objetivo puede verse afectado por limitaciones geográficas ya que pueden existir áreas de difícil cobertura o en las que sea inviable económicamente el desarrollo de infraestructuras tradicionales. En tales entornos se pueden utilizar los protocolos ya implementados en las MANETs para ampliar la cobertura de los sistemas inalámbricos convencionales, tales como GSM (*Global System for Mobile Communications*), WLAN (*Wireless Local Area Networks*) o UMTS (*Universal Mobile Telecommunications System*). En este tipo de escenarios híbridos es necesario una perfecta definición de la interconexión de las redes ad hoc y la red externa o Internet. Esta función de pasarela o *gateway*, puede ser implementada en un nodo estático adicional o integrada en los propios *routers* de acceso. No obstante, es necesario un mecanismo para que los nodos móviles encuentren y establezcan la ruta hacia el *gateway* para así comunicarse con el exterior. Se han desarrollado varios mecanismos que realizan esta función de descubrimiento o localización, los dos más utilizados [1][2] integran este cometido en los *routers* de acceso.

En estos protocolos, el *gateway* incorpora dos funciones principales. En la primera de ellas, se encarga de encaminar adecuadamente el tráfico hacia la red inalámbrica proporcionando así las tareas de encaminamiento ad hoc de las que carece el *router* de acceso. En segundo lugar, difunde por toda la red MANET un mensaje especial que contiene información similar a los mensajes enviados por los *routers* en las redes ad hoc tradicionales pero modificados para que se puedan propagar por toda la red inalámbrica multisalto. Estos mensajes especiales denominados *Modified Router Advertisement* (MRA) son enviados a la red para que los nodos móviles establezcan su dirección IPv6 y, de esta forma, ser alcanzable desde cualquier punto de Internet. Adicionalmente, la recepción de los mensajes MRA permiten que las rutas hacia el *gateway* se actualicen. El mantenimiento de una ruta no válida puede provocar pérdidas y retardos en la red. Cuando un nodo detecta que su ruta hacia el gateway no es válida, envía un mensaje *multicast Modified Router Solicitation (*MRS) para forzar al gateway a que le responda con un mensaje MRA *unicast*.

Los algoritmos de localización de *gateways* se pueden dividir en tres categorías en función de la actuación del gateway: proactivos, reactivos o híbridos. Los algoritmos reactivos sólo buscan las rutas al *gateway* cuando tienen que enviar datos. Por otro lado, en los algoritmos proactivos es el *gateway* el que envía de forma periódica los mensajes MRA para que los nodos actualicen las rutas hacia él. Por último, los métodos híbridos mezclan ambos esquemas comportándose de una manera activa en un área cercana al gateway fuera de la cual los nodos descubren al *gateway* de manera reactiva.

Este artículo se centrará en los algoritmos proactivos. En estos casos el intervalo de tiempo *T* de envío de los mensajes MRA debe elegirse cuidadosamente. Un valor muy bajo de *T* implica el envío de muchos mensajes MRA lo cual puede ser innecesario y en casos extremos saturará a la red. En el caso contrario, un valor alto de *T* puede provocar que los nodos no actualicen de forma adecuada las rutas al *gateway*, se pierdan muchos paquetes y se tengan que iniciar los mecanismos de solicitud de rutas, a través de un mensaje de los mensajes MRS. Por tanto, el intervalo, *T* debe elegirse de tal forma que

se evite el envío innecesario de mensajes de control (ya sean MRA o MRS) que pueden inundar el sistema, pero impidiendo que las rutas necesarias guarden información de encaminamiento erróneas.

En este artículo se presenta un algoritmo para elegir el intervalo basado en tres parámetros: conectividad, estabilidad y número de solicitud de rutas rotas en la red. La topología y la movilidad de los nodos móviles pueden variar estos parámetros muy rápidamente y de forma impredecible en cortos periodos de tiempo. Es por esto, que la incertidumbre asociada a este tipo de redes puede ser modelada mediante un sistema difuso que nos permitirá elegir de forma óptima el instante de envío del siguiente mensaje MRA.

El artículo se divide en las siguientes secciones. En la sección II se describe el estado del arte. En la sección III se presenta el algoritmo de descubrimiento propuesto y que se ha denominado algoritmo difuso para descubrimiento de *gateways*. Las reglas y las funciones del controlador se analizan en la sección IV. El proceso de optimización de los parámetros del controlador se explica en la sección V. Los escenarios de simulación y los patrones de movimiento de los nodos móviles se presentan en la sección VI. Los resultados de las simulaciones se muestran en la sección VIII, para finalizar con las conclusiones en la última sección.

II. ESTADO DEL ARTE.

El uso de la Inteligencia Artificial para mejorar algunas características de las redes MANET ya se ha aplicado en el campo de las comunicaciones. Por ejemplo, en [3] los autores utilizan un protocolo de encaminamiento difuso para el envío de información *multicast*. En ese trabajo se emplean mecanismos de decisión basados en lógica difusa para evaluar el tiempo de vida de las rutas activas y, de esta forma, poder elegir la más adecuada cuando se proceda al envío de datos. Por otra parte, en [4] se incluyen tres algoritmos de encaminamiento en MANET que usan redes neuronales y se comparan entre sí así como con otros protocolos tradicionales.

Respecto a los algoritmos para adaptar los mensajes MRA en la red MANET, uno de los primeros en aparecer es el denominado *Maximal Source Coverage* (MSC) [5]. Aquí, el intervalo de emisión de estos mensajes, *T*, es fijo mientras que el valor del TTL (*Time To Live*) se ajusta dinámicamente de tal forma que se iguala con el valor mínimo para alcanzar a todos los nodos que transmiten datos. El paramétro TTL controla la zona en la que los mensajes MRA se emiten de manera proactiva. Específicamente, estos mensajes representan el número de veces que los mensajes MRA pueden ser retransmitidos.

En [6], los autores desarrollan un nuevo algoritmo que hace uso del número del número de fuentes activas y de *proxies* para ajustar el valor del parámetro TTL y así reducir el número de paquetes de control en la red.

En [7], los autores sugieren que el momento adecuado del envío del MRA depende no sólo del número de usuarios que se comunican con el exterior sino que también hay que tener en cuenta el número de nodos intermedios por los que pasa esa comunicación. Con estos dos parámetros, los autores definen un factor denominado *Regulated Mobility Degree* (RMD) de manera que sólo se enviarían los MRA si este factor sobrepasa un umbral determinado. La principal dificultad de esta propuesta es cómo determinar el umbral a partir de las condiciones de la red.

En [8], se presenta un algoritmo en el que se emplea una estrategia de reenvío de los MRA descentralizada: sólo los nodos que están trasmitiendo paquetes tienen permiso para retransmitir los mensajes MRA. De esta forma, sólo se actualizan las rutas hacia el gateway de las zonas en la que los nodos se están comunicando con el exterior. En aquellas zonas en las cuales no existen fuentes no se retransmiten los mensajes.

En [9], el intervalo *T* se ajusta en función del número de nodos vecinos al *gateway*. Este parámetro está asociado al número total de nodos de la red, de tal forma que si el porcentaje es alto, las rutas serán más cortas. El tener rutas más cortas implica que su tiempo de vida es mayor y, por tanto, existe una menor necesidad de refresco de las mismas. Esta estrategia mejora a las anteriores en situaciones de media y alta tasas de tráfico.

En este trabajo, se han usado para el ajuste del parámetro *T* dos parámetros relacionados con los mensajes MRA recibidos por los *gateways* y retransmitidos por los nodos móviles. Además, se incluye el número de peticiones de nuevas rutas que realizan los nodos móviles de forma reactiva. Con estos parámetros se utiliza un controlador difuso para obtener el siguiente valor del intervalo *T*. Los parámetros del controlador difuso se optimizan mediante un sencillo algoritmo genético.

III. ALGORITMO PROPUESTO

Para mejorar las prestaciones de los algoritmos considerados en el estado del arte anterior, hemos tenido en cuenta tres parámetros: la conectividad, la movilidad y la carga de la red. La influencia conjunta de estos parámetros no está establecida mediante ninguna fórmula empírica ni analítica. Por ejemplo, escenarios con baja conectividad y carga elevada, pueden tener una tasa de pérdidas pequeñas, o puede darse el caso totalmente contrario. Es, por tanto, una buena estrategia utilizar un sistema basado en lógica difusa para optimizar estos tipos de redes en función de estos tres parámetros. El método propuesto consiste en utilizar tres medidas de la red como entradas al sistema difuso:

- La conectividad de la red: definida como el número de nodos vecinos al *gateway*.
- La estabilidad de la red: se obtendrá a partir de los nodos que entran y salen de la cobertura del *gateway*.
- El número de peticiones de ruta por parte de los nodos móviles que envían paquetes de datos. Esta medida proporciona un valor aproximado de la carga de la red.

El factor de conectividad (*cf*) se estima como el número de mensajes MRA recibidos por el *gateway* y que son

retransmitidos por los nodos cercanos al mismo (*NMRA*), dividido por el número medio de estos mensajes, tal y como se refleja en la Ec. 1.

$$cf = \frac{NMRA}{mean(NMRA)} \quad (1)$$

Si *cf* es alto, el número de nodos móviles cercanos al *gateway* es alto. Esto implica que las rutas hacia el *gateway* serán más cortas y tendrán tiempos de vida superiores, con lo que el valor *T* puede ser más elevado [10].

A la entrada del controlador difuso es necesario tener normalizados estos valores, tal y como aparece en la Ec. 2:

$$cf_n = \begin{cases} 1 & si\ cf > kc_2 \\ \dfrac{cf - kc_1}{kc_2 - kc_1} & si\ kc_2 \geq cf \geq kc_1 \\ 0 & si\ cf < kc_1 \end{cases} \quad (2)$$

donde cf_n representa el factor de conectividad normalizado y las variables $\{kc_1, kc_2\}$ son dos umbrales para decidir si este factor es bajo o alto.

El factor de estabilidad (*SF*) es una medida compleja y que se obtiene a partir de la Ec. 3. Este factor analiza los mensajes MRA recibidos durante dos envíos consecutivos.

$$SF = \frac{2 \cdot na - nn}{NMRA(j) + NMRA(j+1)} \quad (3)$$

donde:

- NMRA(j) es el número de mensajes MRA recibidos en el envío *j*.
- *na* (nodos antiguos). Es el número de nodos que permanecen siendo vecinos del *gateway* durante los envíos *j* y *j+1*.
- *nn* (nodos nuevos). Los nodos que están bajo la cobertura del GW en el envío *j+1* y no lo estaban en el envío *j*.

Resultados iniciales con este factor se presentaron en [12]. Si la estabilidad es baja indica que los nodos entran o salen de la cobertura del *gateway* continuamente, con lo cual, el tiempo de refresco de la ruta debe ser menor.

Evidentemente se debe normalizar este valor, de una forma similar a la utilizada en la Ec 2. La nueva variable se denominará SFn o factor de estabilidad normalizado. La normalización se realiza entre dos valores $\{ks_1, ks_2\}$.

El último factor, el de peticiones de rutas *(RRf)* se calcula como el número de mensajes MRS (*NMRS*) dividido por el total de fuentes que se comunican con el exterior (*NSources*), como se indica en la Ec. 4:

$$RRf = \frac{NMRS}{NSources} \quad (4)$$

Si muchas rutas se rompen, este factor será alto y las rutas se deben refrescar con más continuidad.

También se debe normalizar este factor, de una forma similar a la Ec. 2. La nueva variable normalizada o RRf_n representará el factor de peticiones de rutas normalizado y que será uno de los valores de entrada al controlador. Esta normalización se realiza entre dos umbrales $\{kRR_1, KRR_2\}$, por debajo del primer valor se considera que las peticiones son bajas y por encima del segundo que son altas,.

Los tres parámetros anteriores (cf_n, SF_n y RRf_n) constituirán la entrada del controlador difuso. Los seis parámetros necesarios para la normalización y los valores mínimos y máximos de T (kc_1, kc_2, ks_2, ks_1, kRR_1, KRR_2, T_{min}, T_{max}) se obtienen a partir de un proceso de optimización, similar al usado en el entrenamiento de redes neuronales, pero mediante algoritmos genéticos.

IV. Control Difuso

En este trabajo, se van a aplicar técnicas de inteligencia artificial para sintonizar adecuadamente el intervalo de emisión *T* de los mensajes MRA. Como se muestra en la Fig. 1, la pasarela a Internet o IGW(*Internet Gateway*) utiliza los mensajes MRA retransmitidos por los nodos móviles cercanos a él para determinar el instante óptimo de envío del siguiente mensaje de control. La pasarela a Internet calcula tres parámetros: los factores de estabilidad, de conectividad y de peticiones de ruta y, a continuación, los envía a la entrada del controlador difusor.

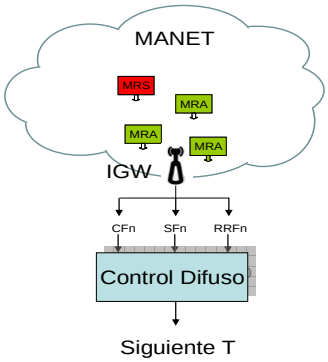


Fig. 1. Esquema de las entradas al controlador difuso.

Los valores a la entrada del controlador ya se encuentran normalizados, la salida del mismo se usará para determinar el siguiente envío del MRA. El valor final del intervalo *T* se obtendrá a partir de la salida anterior y a través de una relación lineal entre dos valores T_{min} y T_{max}, tal y como se observa en la Fig. 2.

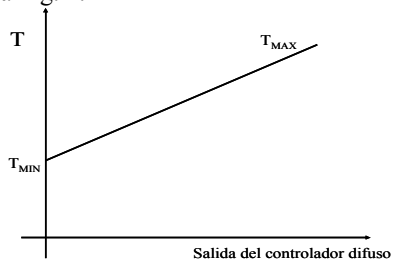


Fig. 2. Relación entre el valor de *T* y la salida del controlador.

Las variables de entrada pueden tener tres valores: bajo, medio y alto. Se han utilizado funciones triangulares para representar los valores anteriores, tal y como se refleja en la Fig. 3. Se han elegido estas funciones por su simplicidad y por su uso extensivo en aplicaciones de tiempo real como es nuestro caso.

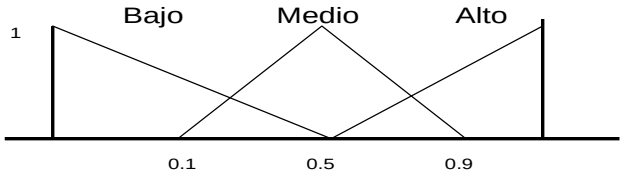


Fig. 3. Funciones de pertenencia para las variables de entrada.

Los términos usados para la selección del siguiente intervalo de emisión son: muy bajo, bajo, moderado, alto y muy alto, tal y como se refleja en la Fig. 4. La salida del controlador estará normalizada en el rango {0,1}.

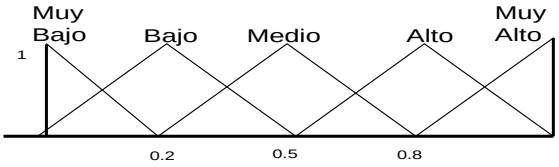


Fig. 4. Funciones de pertenencia para la salida del controlador difuso.

Las reglas que interrelacionan las entradas y las salidas se presentan en la Tabla I. Estas reglas se han elegido gracias a nuestro conocimiento del sistema. Las entradas se combinan a través del operador AND, por ejemplo: si SF_n es alto y CF_n es bajo y RRF_n es bajo, el siguiente intervalo T_n es moderado.

SF_n	CF_n	RRF_n	T_n
ALTO	BAJO	BAJO	MEDIO
ALTO	BAJO	MEDIO	BAJO
ALTO	BAJO	ALTO	MUY BAJO
ALTO	MEDIO	BAJO	ALTO
ALTO	MEDIO	MEDIO	BAJO
ALTO	MEDIO	ALTO	MUY BAJO
ALTO	ALTO	BAJO	MUY ALTO
ALTO	ALTO	MEDIO	ALTO
ALTO	ALTO	ALTO	MEDIO
MEDIO	BAJO	BAJO	MEDIO
MEDIO	BAJO	MEDIO	BAJO
MEDIO	BAJO	ALTO	MUY BAJO
MEDIO	MEDIO	BAJO	MEDIO
MEDIO	MEDIO	MEDIO	BAJO
MEDIO	MEDIO	ALTO	MUY BAJO
MEDIO	ALTO	BAJO	ALTO
MEDIO	ALTO	MEDIO	BAJO
MEDIO	ALTO	ALTO	MUY BAJO
BAJO	BAJO	BAJO	BAJO
BAJO	BAJO	MEDIO	MUY BAJO
BAJO	BAJO	ALTO	MUY BAJO
BAJO	MEDIO	BAJO	BAJO
BAJO	MEDIO	MEDIO	BAJO
BAJO	MEDIO	ALTO	MUY BAJO
BAJO	ALTO	BAJO	MEDIO
BAJO	ALTO	MEDIO	BAJO
BAJO	ALTO	ALTO	BAJO

Tabla 1. Base de reglas del controlador difuso.

V. PROCESO DE OPTIMIZACIÓN.

En esta sección se explica brevemente el proceso de optimización que se utiliza para realizar la normalización de los parámetros de entrada al controlador difuso. En el proceso de optimización se usa un algoritmo genético (AG). Los algoritmos genéticos se utilizan a menudo para encontrar soluciones a problemas sin solución analítica o del tipo np. Un algoritmo genético codifica cada solución a través de un individuo o cromosoma. El algoritmo mantiene una población de individuos de tal forma que en la siguiente iteración se cambian ciertos individuos usando los mejores cromosomas de esa iteración. La calidad del individuo se mide a través de una función de evaluación. Este valor nos sirve para comparar unos individuos con otros.

Para codificar el problema que nos ocupa, cada individuo estará formado por un vector de los siguientes ocho elementos: $\{kc_2, kc_1, ks_2, ks_1, krr_2, krr_1, T_{min}, T_{max}\}$.

El proceso de optimización es similar al del entrenamiento de una red neuronal. En concreto usamos tres conjuntos distintos:

- En primer lugar, seleccionamos 10 escenarios que formarán el conjunto de optimización. El AG utiliza este conjunto para encontrar la mejor solución posible.
- Con la solución anterior, se realiza un test con 10 nuevos escenarios. A este conjunto se le llama conjunto de validación.
- Si el resultado no es el adecuado volvemos al primer paso.
- Finalmente, los resultados finales se obtienen con 50 nuevos escenarios denominados conjunto de validación.

El proceso anterior se resume en la Fig. 5:

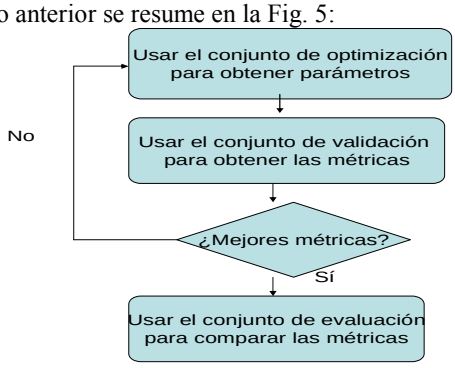


Fig. 5. Optimización con AG.

La función de evaluación que se usa contiene los tres parámetros más importantes que percibe un usuario cuando realiza una comunicación. Estos parámetros son los siguientes:

- La tasa de pérdidas o *PLR* (*packet loss rate*): definida como el número de paquetes pérdidos sobre el total de paquetes transmitidos.
- El retardo: el valor medio del tiempo que tardan los paquetes recibidos en llegar a su destino.
- La carga normalizada o *NRO* (*Normalized Routing Overhead*): definida como el número total de paquetes de control dividido por el número total de paquetes recibidos. Para el cálculo de este parámetro, cada retransmisión de un paquete de control se considera como uno nuevo.

Estos parámetros proporcionan una estimación de las condiciones de la red. La tasa de pérdidas y el retardo son los aspectos más importantes a considerar desde el punto de vista de los usuarios, mientras que la carga normalizada es una medida importante que nos informa sobre el consumo de los dispositivos, estimación muy importante en estos dispositivos con baterías tan limitadas.

La función de evaluación considerada para combinar los distintos individuos es la siguiente:

$$f = \frac{delay}{\min(delay)} + \delta(delay > \min(delay)) + \frac{plr}{\min(plr)} + \cdots \\ \cdots + \delta(plr > \min(plr)) + \frac{NRO}{\min(NRO)} + \delta(NRO > \min(NRO)) \quad (5)$$

donde $\delta(u)$ es la función escalón.

En la función anterior hemos sumado las aportaciones de los tres parámetros usados para comprobar la eficacia de cada uno de los algoritmos encontrados en la literatura. Cada uno de estos parámetros se divide por el valor mínimo del mismo parámetro, considerando todos los algoritmos evaluados. Además, como queremos disminuir por igual los tres parámetros, incorporamos la función escalón para no encontrar mínimos locales de un solo parámetro.

VI. ESCENARIO Y PATRONES DE MOVILIDAD.

En los últimos años la evaluación de distintos algoritmos en la redes MANET se realizaban con patrones de movilidad limitados como puede ser el *Random Waypont*. Sin embargo recientemente se empiezan a usar otros patrones en los cuales se incluyen trazas lo más parecidas a la realidad posibles. Siguiendo esta argumentación se ha usado para la caracterización de los algoritmos el *Time-variant Community Mobility Model* (TCMM) [12]. EL TCMM es un modelo realista que se obtiene a través de trazas de LAN (*Local Area Network*) inalámbricas. Los autores incorporan en este modelo dos características importantes: las preferencias de cada usuario y las reapariciones periódicas. Para incluir estos parámetros define las denominadas comunidades, que son aquellas zonas visitadas frecuentemente por los nodos. En las simulaciones se han usado dos comunidades emplazadas aleatoriamente en el escenario de simulación.

El escenario elegido cuenta con un único *gateway* situado en el centro del mismo. El resto de las características se presentan en la Tabla 2:

Rango de Transmisión	250 m
Protocolo Ad hoc	AODV (Ad Hoc On Demand Distance Vector Routing) Local repair disabled
Capa de enlace	Detección de la capa de enlace habilitada 802.11 RTS/CTS habilitada
Patrón de movilidad	Máxima velocidad: {2,35} m/s Tiempo de pausa: 10 s
Fuente de tráfico	10 fuentes (CBR) Tamaño del paquete: 128 bytes Tasas: 5, 10, 15 paquetes/s
Nodos	50
Número de comunidades	2
Dimensiones	1500 x 300 m^2
Localización del IGW	(750 ,150) m

Tabla 2. Características del escenario de simulación

VII. RESULTADOS

Los distintos algoritmos adaptativos se han implementado y simulado con el *Network Simulator* [13] (en concreto la versión 2.31 con el sistema operativo Linux). Las simulaciones son de 1000 segundos para tener resultados significativos [14]. Al estar interesados sólo en los valores finales hemos eliminado de los resultados los 100 primeros segundos, considerándolos como transitorio.

A través del proceso de optimización con AG se obtuvo en primer lugar los valores óptimos del vector {kc_2, kc_1, ks_2, ks_1, krr_2, krr_1, T_{min}, T_{max}} que se encuentran en la Tabla 3:

Kc_2	kc_1	ks_2	ks_1	krr_2	krr_1	T_{min}	T_{max}
0.80	0.20	-0.60	0.00	0.50	0.20	4.95	39.95

Tabla 3. Valores óptimos para el controlador difuso.

A continuación presentamos los resultados para nuestro algoritmo *Adaptive Genetic Fuzzy Discovery Gateway* (AGF) (en la última fila de las tablas siguientes), y lo comparamos con otros algoritmos como son: MSC [5], el *Low Overhead and Scalable Proxied* (LOSP) [6], the RMD [7], the ADD [8] and the Adaptive Gateway (AGW) [9] descritos en la Sección II. Los parámetros necesarios para cada uno de los algoritmos anteriores se han obtenido de sus respectivas referencias. Cada valor en la tabla es el resultado de cincuenta simulaciones (del denominado conjunto de validación). En las tablas se incluyen los resultados para un tasa de paquetes baja (5 paquetes/s), media (10 paquetes/s) y alta (15 paquetes/s).

Algoritmo	***Retardo(s)***	***PLR***	***NRO***
LOSP	*0.0586*	*0.0094*	*0.3710*
RMD	*0.0537*	*0.0081*	*0.3100*

AGW	*0.0488*	*0.0076*	*0.2944*
ADD	*0.0546*	*0.0087*	*0.3196*
MSC	*0.0534*	*0.0094*	*0.3759*
AGF	***0.0492***	***0.0070***	***0.2632***

Tabla 4. Resultados para baja tasa de datos.

Algoritmo	***Retardo (s)***	***PLR***	***NRO***
LOSP	*0.0547*	*0.0100*	*0.2697*
RMD	*0.0530*	*0.0091*	*0.2399*
AGW	*0.0499*	*0.0079*	*0.2153*
ADD	*0.0534*	*0.0095*	*0.2417*
MSC	*0.0545*	*0.0096*	*0.2771*
AGF	***0.0491***	***0.0075***	***0.2005***

Tabla 5. Resultados para tasa media de datos.

Algoritmo	***Retardo (s)***	***PlR***	***NRO***
LOSP	*0.1826*	*0.1303*	*1.0166*
RMD	*0.1777*	*0.1289*	*0.9954*
AGW	*0.1727*	*0.1236*	*0.9550*
ADD	*0.1784*	*0.1283*	*0.9972*
MSC	*0.1802*	*0.1308*	*1.0233*
AGF	***0.1683***	***0.1222***	***0.9453***

Tabla 6. Resultados para tasa alta de datos.

Como se observa en las tablas AGF es el mejor algoritmo para todos los casos considerados. Por otra parte LOSP presenta menor NRO que MSC pero con un ligero incremento en *delay* y plr sobre MSC. El segundo de los algoritmos es el AGW, pero siempre con valores peores que el AGF.

VIII. CONCLUSIONES

La incertidumbre inherente a las redes MANET, tanto en estabilidad de los enlaces como en el movimiento relativo de unos nodos con otros, nos han llevado a usar un controlador difuso para mejorar las características de este tipo de redes. El controlador difuso se ha optimizado mediante un algoritmo genético. El controlador utiliza tres parámetros importantes de este tipo de redes: las peticiones de nuevas rutas, la conectividad y la estabilidad de la misma. El controlador difuso sólo se usa en la pasarela a Internet, con lo cual los nodos móviles no tienen que llevar a cabo nuevas funciones ni cambiar el software de los mismos, lo cual podría implicar una menor duración de la batería y de la capacidad de almacenamiento de los mismos. El nuevo esquema mejora significativamente el proceso de conexión de la red MANET con el exterior respecto a otros mecanismos que aparecen en la literatura. Esta mejora incluye tanto el retardo, las pérdidas como la carga de la red.

Como trabajo futuro, el algoritmo AGF se usará con otros escenarios y patrones de movilidad, incluyendo un nuevo proceso de optimización de parámetros *offline* como puede ser la optimización de particulas o *Particle Swarm Optimization* (PSO).

REFERENCIAS

[1] R. Wakikawa, J.T. Malinen, C.E. Perkins, A. Nilsson and A.J. Tuominen: Global Connectivity for IPv6 Mobile Ad Hoc Networks. Internet Engineering Task Force, Internet Draft (work in progress)

[2] C. Jelger, T. Noel and A. Frey: Gateway and Address Autoconfiguration for IPv6 Ad Hoc Networks. Internet-Draft draft-jelger-manet-Gateway-autoconf-v6-02.txt (Abril 2004)

[3] B.L. Su, M.S. Wang and Y.M. Huang, "Fuzzy logic weighted multi-criteria of dynamic route lifetime for reliable multicast routing in ad hoc networks", in Expert Systems With Applications, vol. 35, nº 1-2, Elsevier, 2008, pp. 476-484.

[4] N.H. Saeed, M.F. Abbod, H.S. Al-Raweshidy. "Modeling MANET Utilizing Artificial Intelligent", in Second UKSIM European Symposium on Computer Modeling and Simulation, 2008. EMS '08. pp. 117-122. Liverpool. U.K. (Abril 2008)

[5] P.M. Ruiz and A.F. Gomez-Skarmeta: "Maximal Source Coverage Adaptive Gateway Discovery for Hybrid Ad Hoc Networks", Lecture Notes in Computer Science, vol. 3158, pp. 28-41 (Julio 2004)

[6] F.J. Ros, P.M. Ruiz, "Low Overhead and Scalable Proxied Adaptive Gateway Discovery for Mobile Ad Hoc Networks", The Third IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2006), pp. 226-235, Vancouver, Canada (Octubre 2006)

[7] V. Rakeshkumar and M. Misra: "An Efficient Mechanism for Connecting MANET and Internet through Complete Adaptive Gateway Discovery." In: First International Conference on Communication System Software and Middleware (COMSWARE2006). pp. 1-5, New Delhi, India (Enero 2006)

[8] U. Javaid, F. Rasheed, D.-E. Meddour, T. Ahmed, T.,"Adaptive Distributed Gateway Discovery in Hybrid Wireless Networks". In: Wireless Communications and Networking Conference, 2008. WCNC 2008. Las Vegas. USA (Marzo-Abril 2008)

[9] F.D. Trujillo, A. Triviño, E. Casilari, A. Díaz-Estrella and A.J. Yuste: An adaptive Gateway discovery in hybrid MANETs. In: Fourth EuroFGI Workshop on Wireless and Mobility, pp. 55-58. Barcelona, Spain (Enero 2008)

[10] A. Triviño-Cabrera, J. García-de-la-Nava, E. Casilari and F.J. González-Cañete: An Analytical Model to Estimate Path-Duration in MANETs. In: 9th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM2006). Torremolinos, Spain (Octubre 2006)

[11] F.D. Trujillo Aguilera, A.J. Yuste Delgado, A. Trivino Cabrera, E. Casilari Perez, and A. Diaz Estrella. "Interconnecting MANET and the Internet. A mobility approach". In: The 33rd Annual IEEE Conference on Local Computer Networks (LCN08). Montreal, Canada (Octubre 2008)

[12] W. Hsu, T. Spyropoulos, K. Psounis, A. Helmy. In: "Modeling Time-variant User Mobility in Wireless Mobile Networks," INFOCOM 2007, May 2007, pp 758-766. Anchorage, USA (Mayo 2007)

[13] Network simulator, http://www.isi.edu/nsnam/ns

[14] E. Casilari and A. Triviño-Cabrera: A practical study of the Random Waypoint mobility model in simulations of ad hoc networks. In: 19th International Teletraffic Congress (ITC'19). pp. 115-124, Beijing , China (Agosto 2005)

Influencia de la Directividad en el Rendimiento de Protocolos Ad-hoc de Enrutamiento para Redes Multi-Hop Cellular

Baldomero Coll Perales y Javier Gozálvez Sempere

Ubiquitous Wireless Communications Research Laboratory
Uwicore, http://www.uwicore.umh.es
Universidad Miguel Hernández de Elche
Avda de la Universidad, s/n, 03202 Elche, España
Mail: bcoll@umh.es, j.gozalvez@umh.es, tlf:+34 966658955

***Resumen*—El rendimiento de los algoritmos de enrutamiento para redes inalámbricas multi-hop tiene una alta dependencia con la función de costes establecida para definir la ruta óptima. En las redes Multi-Hop Cellular, la transmisión entre terminal móvil y estación base se realiza utilizando nodos retransmisores. En este contexto, este artículo propone la utilización de la direccionalidad hacia el destino como parámetro a considerar en la selección de la ruta entre origen y destino para algoritmos de enrutamiento multi-hop en redes MCN. Los resultados obtenidos demuestran que la consideración de la direccionalidad incrementa la eficiencia del protocolo de enrutamiento al reducir de forma notable la señalización y energía consumida en el proceso de enrutamiento.**

***Palabras clave*—redes MCN, enrutamiento ad-hoc, función de costes, direccionalidad al destino**

I. INTRODUCCION

Según define la Unión Internacional de las Telecomunicaciones (UIT) [1], la futura generación de redes celulares (4G) deberá proveer de altas tasas de transmisión en amplias zonas y de forma homogénea. Las actuales redes celulares, denominadas también de único salto (*Single-hop Cellular Network*, SCN) debido a que la comunicación se realiza de un modo directo entre el terminal móvil y la estación base (*Base Station*, BS), son incapaces de ofrecer altas tasas de transmisión de forma homogénea en su celda de cobertura, debido a la creciente atenuación de la señal a medida que aumenta la distancia entre el terminal móvil y la BS, y la dependencia de la tasa de transmisión con dicha distancia o atenuación de la señal. La consecución de los objetivos establecidos por la UIT para la 4G pasan por la instalación de un mayor número de BS, y/o hacer uso de la tecnología ad-hoc. La instalación de un mayor número de BS cuenta con el consecuente gasto económico, además del creciente rechazo social ante la instalación de elementos radiantes. La solución que parece estar tomando ventaja consiste en alcanzar a la BS a través de múltiples saltos, dando lugar a lo que se conoce como redes MCN (*Multi-hop Cellular Network*) [2]-[3]. En las redes MCN, la comunicación entre el terminal móvil y la estación base no se realiza de forma directa sino a través de nodos de comunicación intermedios que retransmiten la información del nodo fuente hacia el nodo destino. La integración de las comunicaciones ad-hoc en las redes celulares permite la reducción de la distancia entre cada uno de los saltos participantes en la comunicación entre origen y destino, con lo que se mejoran las condiciones de propagación, el nivel de señal recibida, y consecuentemente las tasas de transmisión.

Dentro de las redes MCN pueden diferenciarse entre las que emplean retransmisores fijos (MCN-Fixed Relay, MCN-FR), y las que usan a los propios terminales móviles como retransmisores (MCN-Mobile Relay, MCN-MR). Si bien la complejidad del diseño y desarrollo de las redes MCN-FR es notablemente más baja que la de las redes MCN-MR, presentan el inconveniente de un mayor coste económico al requerir de la instalación de nuevos nodos retransmisores, y esto en un entorno de creciente rechazo a las antenas de telefonía móvil y comunicaciones inalámbricas. Por el contrario, las redes MCN-MR explotan la capacidad de los terminales móviles para optimizar el rendimiento de las redes de comunicaciones móviles a través de protocolos multi-hop, que integran los sistemas celulares y ad-hoc, y sin la necesidad de desplegar nuevos elementos radiantes. Sin embargo, el adecuado desarrollo y despliegue de redes MCN-MR requiere todavía de superar importantes retos tecnológicos, como el de la selección de rutas entre los nodos origen y destino eficientes, robustas, y que no requieran de un alto consumo energético por parte de los nodos retransmisores. En este contexto, el presente artículo se centra en el estudio y desarrollo de algoritmos de enrutamiento en redes MCN-MR que permitan seleccionar las posibles rutas entre los nodos origen y destino en base a unos criterios preestablecidos que explotan las características de las redes MCN-MR. En concreto, el artículo propone la consideración de la direccionalidad en el establecimiento de rutas multi-hop para así optimizar el rendimiento de las redes MCN-MR, reduciendo también la señalización requerida para el descubrimiento de rutas.

II. PROTOCOLO DE ENRUTAMIENTO AODV

A. Funcionamiento básico de AODV

El trabajo desarrollado se basa en el protocolo de enrutamiento AODV (*Ad-Hoc On-Demand Distance Vector*) [4]. AODV es un protocolo de enrutamiento reactivo que

únicamente realiza la búsqueda de la ruta entre fuente (S) y destino (D) cuando el nodo fuente *S* tiene información que transmitir. El proceso de creación de ruta entre los nodos *S* y *D* se basa, principalmente, en dos mecanismos: la petición de ruta por parte del nodo *S*, y la respuesta del nodo *D*. Cuando el nodo *S* tiene información que transmitir y no conoce el camino para llegar a *D*, envía en modo broadcast un paquete de petición de ruta (*Route REQuest*, *RREQ*) que es difundido por la red a través de las retransmisiones de los nodos intermedios. Cuando el *RREQ* alcanza al nodo *S*, éste responde con un paquete unicast (*Route REPly, RREP*) confirmando la creación de la ruta. La recepción de los paquetes *RREQ* y *RREP* permite, a los nodos intermedios, conocer al nodo que les precede en el camino hacia *S* y *D* respectivamente, y por lo tanto, conocer a quien deben dirigir la información para alcanzar al nodo destino.

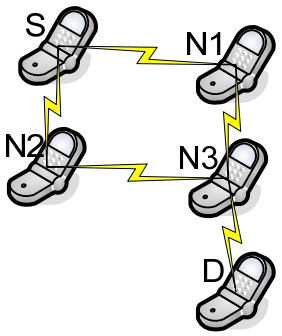


Figura 1. Ejemplo de funcionamiento de AODV

Supongamos que, como se muestra en la Figura 1, únicamente los nodos N_1 y N_2 se encuentran dentro del radio de cobertura de *S*. Ante la llegada del *RREQ*, tanto N_1 como N_2 comprobarán que no son el destinatario del paquete y, tras almacenar el identificador del nodo del que han recibido el *RREQ*, reenvían en modo broadcast el paquete. El envío realizado por N_1 y N_2 provoca dos nuevos eventos. Por un lado, el paquete *RREQ* vuelve a llegar a *S*, que tras reconocer el paquete lo desechará. Por otro lado, tanto el *RREQ* de N_1 como el de N_2 alcanzarán a N_3 (suponemos que N_1 y N_2 no se encuentran dentro de sus respectivos radios de cobertura). En la versión original de AODV, N_3 únicamente reenvía el primero de los *RREQ* que escucha, el resto de *RREQ* originados en el mismo intento de búsqueda de ruta (reconocidos con un número de secuencia) son desechados. Supongamos que el *RREQ* enviado por N_1 alcanza antes a N_3 que el de N_2. En ese caso, N_3 almacenará en su tabla de ruta que para llegar hasta *S* debe mandar el paquete por N_1. Cuando a N_3 llegue el *RREQ* enviado por N_2 lo descartará tras reconocer que posee el mismo número de secuencia que el que recibió anteriormente de N_1. N_3, por no ser el destinatario del paquete, vuelve a reenviar el *RREQ*, dándose de nuevo otras dos circunstancias. Por un lado el *RREQ* llega a N_1 y N_2 que descartan el paquete reconociendo el número de secuencia, y por otro lado, el *RREQ* alcanza al nodo *D*. El nodo *D* almacena en su tabla de ruta que para alcanzar al nodo *S* debe enviar el paquete por N_3, además de verificar que él es el destinatario del *RREQ* y generar un paquete de respuesta *RREP*. Debido a que *D* posee una ruta para alcanzar a *S* (la creada con la recepción del *RREQ*), el envío del *RREP* se realiza en modo unicast. De este modo, el paquete *RREP* pasará de *D* a N_3, de N_3 a N_1 y de N_1 a *S*. La llegada del *RREP* permite a los nodos intermedios conocer la identificación del nodo que les precede en el camino hacia *D*, y por tanto conocer a quien deben enviar la información si se desea alcanzar a *D*. Así, por ejemplo, cuando N_1 recibe el *RREP* de N_3, provoca que N_1 almacene en su tabla de rutas que para llegar a *D* debe enviarle el paquete a N_3. Cuando el *RREP* alcanza a *S*, el proceso de creación de ruta concluye y da comienzo la transmisión multi-hop de la información.

B. Modificaciones del algoritmo AODV

En la versión original de AODV, la ruta escogida entre los nodos *S* y *D* es aquella que posee una menor latencia, lo cual suele coincidir con la ruta que realiza un menor número de saltos. Esto implica que no se tiene en cuenta el estado de la red a la hora de construir la ruta. Por ello, el presente artículo implementa la modificación del protocolo AODV propuesta en el estándar 802.11s, y en concreto el procedimiento de aceptación de paquetes de búsqueda de ruta propuesto en el mecanismo on-demand de enrutamiento [5]. 802.11s es un estándar creado para permitir la formación de redes mesh inalámbricas. Sus modificaciones respecto a 802.11 están centradas principalmente a nivel MAC, desarrollando funcionalidades como el descubrimiento de la red mesh, autentificación, gestión de enlaces, selección de canal, seguridad, interworking y selección de camino entre otros.

La modificación de AODV propuesta permite que la ruta de menor latencia se siga construyendo, pero ante la llegada de otro *RREQ* con el mismo número de secuencia, la modificación implementada verifica si el coste del nuevo camino propuesto es mejor o peor que el almacenado. Para realizar el cálculo del coste del camino se emplea la función de costes que se haya definido e implementado.

Para llevar a cabo la nueva implementación es necesario incorporar en el paquete *RREQ* un nuevo campo, en el que está almacenado el coste acumulativo de la ruta. Haciendo uso del ejemplo visto en la Figura 1, cuando N_2 y N_1 reciban el *RREQ* proveniente de *S* ambos calculan el coste del enlace con *S*. El valor del coste del enlace es almacenado tanto en la tabla de ruta como en el *RREQ*. Igual que ocurría en AODV, el *RREQ* retransmitido por N_1 alcanza a N_3 antes que el retransmitido por N_2. Ante la llegada del *RREQ*, N_3 calcula el coste del enlace con N_1 y, al tratarse de un nuevo número de secuencia, almacena en la tabla de ruta la suma del coste del enlace con N_1 y el valor del campo de coste acumulativo que contiene el paquete *RREQ*. N_3, tras actualizar el paquete *RREQ* (coste acumulativo, etc.), lo retransmite. La modificación sobre AODV afecta cuando el *RREQ* proveniente de N_2 alcanza N_3. En la versión original de AODV sería descartado directamente. Por el contrario en la versión modificada de AODV, N_3 calcula el coste del enlace con N_2 y tras sumarlo al coste acumulativo que contiene el *RREQ*, comprueba si es mayor o menor que el que ya tiene almacenado en su tabla de ruta. En caso de que sea mayor, el

paquete es descartado, pero si el coste acumulativo de la nueva ruta es menor, el paquete será aceptado, la tabla de ruta será actualizada, y se ejecutará un nuevo reenvío del *RREQ* (Figura 2). De este modo es posible que a la estación *D* llegue más de un *RREQ*.

La principal ventaja de la modificación propuesta por 802.11s es que el *RREQ* que contienen la ruta de menor latencia ($RREQ_1$) es el primero en llegar a *D*, y al que inmediatamente se responde con el *RREP*. Pero además, se construye la ruta que mejor se adapta a las condiciones impuestas en la función de costes, ruta creada a partir del *RREQ* que minimiza la función de costes definida para el protocolo de enrutamiento ($RREQ_{min}$). Otra propiedad de la modificación es que los cambios realizados en las tablas de rutas por $RREQ_{min}$ son válidos para el *RREP* que generara $RREQ_1$. Para el caso de la Figura 2, supongamos que en esas condiciones es generado el *RREP* que provoca $RREQ_1$. Cuando el *RREP* alcance a N_3, en la tabla de ruta se informará que para llegar a *S* el siguiente salto es N_2, a pesar de que la ruta creada por $RREQ_1$ se había construido por N_1.

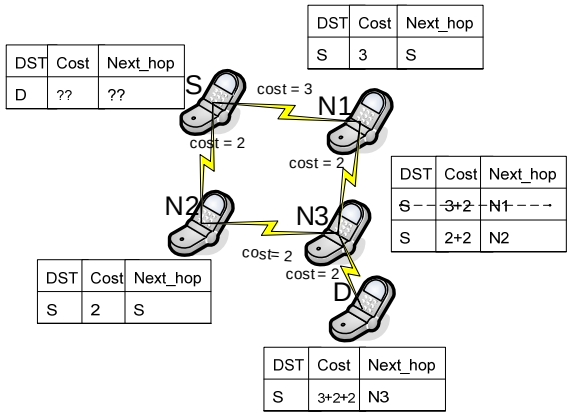


Figura 2. Ejemplo de funcionamiento de la modificación de AODV propuesta en el estándar 802.11s

III. PROTOCOLOS DE ENRUTAMIENTO MULTI-HOP

A. Introducción

Diferentes protocolos de enrutamiento con variadas funciones de coste han sido propuestos para redes inalámbricas, y en menor medida para redes MCN. Las primeras propuestas [4]-[6], influenciadas por los mecanismo de enrutamiento para redes cableadas, consideraban únicamente el número de saltos entre las estaciones origen *S* y destino *D* como criterio para el establecimiento de la ruta. El menor número de saltos suele coincidir con la ruta de menor latencia, razón por la cual, a pesar de su sencillez, sigue siendo el número de saltos un parámetro ampliamente empleado en protocolos de enrutamiento multi-hop.

Desde entonces, numerosos parámetros han sido propuestos con la finalidad de tener en cuenta el estado de la red durante el proceso de establecimiento de la ruta, dependiendo su relevancia de las características de la red. Así, en las redes inalámbricas, aspectos como la energía [7], el nivel de congestión [8], la tasa de paquetes erróneamente recibida o PER [9], y la tasa de transmisión o *throughput* [10], entre otros parámetros, pueden presentar un notable impacto en el eficiente establecimiento de enlaces multi-hop, y por lo tanto en el rendimiento de los protocolos de enrutamiento. Además de la definición de nuevas funciones de coste para la identificación de rutas multi-hop, investigadores han propuesto también modificaciones en los protocolos de enrutamiento más aceptados como AODV. Por ejemplo, en [11], los autores desarrollan diferentes procedimientos a seguir durante el proceso de búsqueda de ruta. En el primero de ellos, los nodos intermedios descartan cualquier paquete de búsqueda de ruta si un paquete procedente de la misma fuente ya ha sido escuchado. Por otro lado, el segundo algoritmo establece un tiempo de espera durante el cual son aceptados todos los paquetes de búsqueda de ruta, siempre y cuando el camino recorrido por el paquete posea un coste acumulativo menor.

B. Mecanismo Multi-Métricas

Con el fin de evaluar la propuesta realizada en este trabajo, es decir la consideración de la direccionalidad en el proceso de enrutamiento en las redes MCN-MR, el presente trabajo ha implementado como algoritmo base con el que realizar las comparaciones el propuesto en [12]. De igual modo que en muchas otras propuestas, [12] presenta una función de costes lineal, en la que cada uno de los parámetros que intervienen en ella están ponderados para modificar su influencia en el computo global de la función de costes. La elección del número de saltos, la congestión del canal y la energía permiten la creación de rutas teniendo en consideración las principales limitaciones de las redes inalámbricas. A pesar de usar los parámetros propuestos por [12] para la función de costes, se han realizado modificaciones en la implementación al variar la tecnología sobre la cual el protocolo ha sido implementado. Por ejemplo, en el caso de la energía, en la función de costes presentada en [12], debido a la implementación de un mecanismo de control de potencia, la función de costes trata de minimizar el coste energético a través de la reducción de la potencia de transmisión. La implementación presentada en este artículo, al no considerar mecanismos de control de potencia, considera el factor energético intentando maximizar el tiempo de vida de las baterías.

La función de costes definida en [12] permite crear caminos entre *S* y *D* caracterizando cada uno de los enlaces que posee un nodo, y asignándoles un coste. De este modo, en la construcción de la ruta se busca el camino de mínimo coste. Existen numerosos parámetros a considerar en la función de costes, pero ha de tenerse en cuenta el compromiso entre los beneficios de una compleja función de costes y el propio coste computacional que ello conlleva. La función de costes definida en [12] e implementada en el presente trabajo para fines comparativos está modelada por la siguiente ecuación:

$$\text{cost} = \alpha_1 \cdot 1 + \alpha_2 \cdot load + \alpha_3 \cdot energy \quad (1)$$

La ecuación (1) es calculada por cada uno de los nodos al recibir el *RREQ*. El valor de *cost* representa el coste del enlace entre el nodo que envía y el que recibe el *RREQ*. En ella, "1" hace referencia al número de saltos, *load* representa la congestión del canal y *energy* denota la energía consumida por el nodo. A continuación se realiza una explicación más detallada de cada uno de los parámetros y del modo en el que son calculados.

- **"1"**: Representa el salto necesario para llegar desde el nodo que envía el *RREQ* hasta el nodo que lo recibe.
- ***load***: Para realizar la medida de la congestión del canal se utiliza el envío periódico de los mensajes de beacon. Puesto que el intervalo entre mensajes de beacon es conocido, es posible determinar el retraso sufrido por el mensaje. Este retraso es causa, principalmente, de la contienda por el canal y/o de una acumulación en el buffer de envío de paquetes. La ecuación (2) muestra como es calculado el intervalo entre los mensajes de beacon. En (2), *NOW* representa el instante actual, T_{last_beacon} el instante en el que se recibió el último mensaje de beacon, y T_{beacon} el periodo de beacon. La ecuación (3) pretende que la medida de la congestión del canal sea consecuente con el historial del enlace. El parámetro β permite dar mayor o menor importancia al último valor del intervalo de beacon calculado.

$$\text{intvl}=\frac{\text{NOW}-T_{\text{last\_beacon}}-T_{\text{beacon}}}{T_{\text{beacon}}};\ 0 \le \text{intvl} \le 1 \qquad (2)$$

$$load = (1-\beta)\cdot load_{\text{old}} + \beta \cdot \text{intvl} \qquad (3)$$

- ***energy***: El parámetro *energy* representa la energía consumida por un nodo. Este parámetro es incorporado en la función de costes con la finalidad de maximizar el tiempo de vida de las baterías. Como puede verse en la ecuación (4), el valor de *energy* es calculado mediante la resta de la energía inicial (E_{init}) y la energía actual (E_{now}).

$$energy = E_{init} - E_{now} \qquad (4)$$

Como puede apreciarse en la ecuación (1), tanto el número de saltos, como la congestión del canal y la energía están ponderados por unas constantes α_i. Con el fin de que cada uno de los parámetros tenga igual influencia dentro de la función de costes, los valores de α_i han de ser los siguientes:

$$(\alpha_1, \alpha_2, \alpha_3) = (1, 1, \frac{1}{E_{init}}) \qquad (5)$$

Para medir la influencia de la direccionalidad hacia el destino en algoritmos de enrutamiento ad-hoc han sido implementadas diferentes versiones del algoritmo de selección de ruta. Los algoritmos propuestos utilizan la función de costes presentada en esta sección para escoger la ruta de mínimo coste entre todas las rutas que alcancen al destino.

C. *Consideración de la direccionalidad en los protocolos de enrutamiento para redes MCN-MR*

Las redes MCN-MR llevan a cabo la comunicación entre un terminal móvil y una estación base empleando nodos retransmisores móviles. En el caso del enlace ascendente o *uplink*, en el que se centra la presente investigación, es factible considerar, incluso actualmente, que el terminal móvil origen pudiera conocer la ubicación fija de la estación base destino de los datos; la ubicación de las estaciones base de telefonía móvil en España está actualmente disponible al público en la web del Ministerio de Industria, Turismo y Comercio (www.mityc.es). En el caso del tráfico descendente o *downlink*, la posición del terminal móvil destino de la información podría estar disponible de forma precisa a través del sistema GPS o Galileo, aunque también podría estimarse de forma más o menos precisa a través de técnicas de localización geométrica en redes de telefonía móvil. En este contexto, el presente artículo propone la consideración de la direccionalidad desde el nodo fuente de la información hasta el nodo destino con el fin de mejorar el enrutamiento en redes inalámbricas multi-hop, sobre todo reduciendo la señalización, congestión y consumo energético que caracterizan a la mayoría de protocolos de enrutamiento multi-hop disponibles en la actualidad.

IV. PLATAFORMA DE SIMULACIÓN

El presente estudio ha sido desarrollado empleando la plataforma de simulación ns2 (*Network Simulator v.2*) [13] con el fin de modelar la comunicación entre los distintos nodos del sistema. ns2 es un simulador de eventos discretos y de código abierto ampliamente utilizado en la investigación de redes de comunicación tanto cableadas como inalámbricas. El alto grado de aceptación que posee ns2 en distintos ámbitos de investigación, incluido el estudio de redes *ad-hoc* de comunicaciones, permitirá contrastar, comparar y verificar los resultados obtenidos.

Las características del escenario de simulación en el que será evaluado el rendimiento de los algoritmos propuestos en este artículo vienen resumidas en la Tabla 1. Sobre un escenario tipo Manhattan de dimensiones 2250m x 2250m, diferentes densidades de nodos se desplazan siguiendo un modelo de movilidad *Random Walk Obstacle* [14]. Dicho modelo de movilidad establece inicialmente una posición aleatoria de los nodos. Cuando da comienzo la simulación, los nodos se desplazan con una velocidad constante hasta alcanzar una intersección. La probabilidad de cambiar de dirección en la intersección es del 25%, es decir, el nodo puede girar a la derecha, girar a la izquierda, seguir recto o retroceder con la misma probabilidad. Para dar mayor realismo al escenario de simulación, existen dos vías principales que se cruzan en el centro del escenario, donde está situada la BS. La Figura 3 ofrece una visión del entorno de simulación para una densidad de 500 nodos. La anchura de los edificios es de 225 m y la de las calles de 25 m.

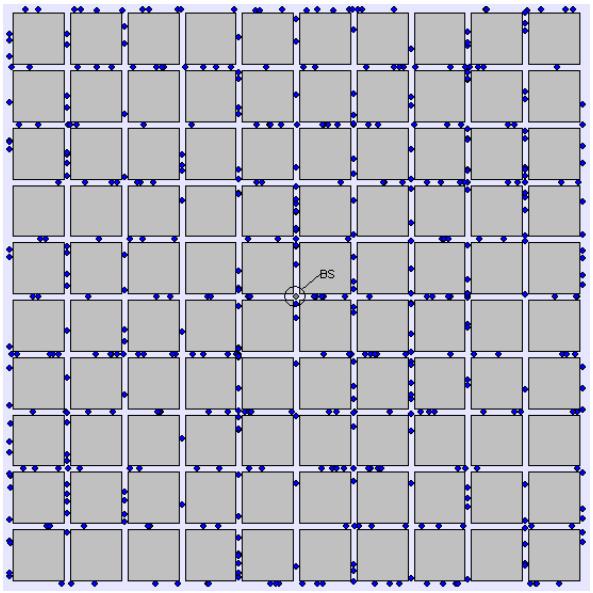


Figura 3. Escenario de simulación

La tecnología de acceso radio empleada por los nodos es 802.11a, la cual trabaja en la banda de frecuencias de 5.8GHz. El modelo de propagación considerado para modelar las pérdidas que sufre la señal radio al propagarse entre transmisor y receptor es un modelo determinista que considera el efecto del *pathloss*, diferenciando la atenuación de la señal en base a la distancia entre nodos de comunicación, y la existencia o no de condiciones de visión directa (LOS - Line-of-Sight) y NLOS (Non Line-of-Sight). En un entorno urbano, como el simulado en el presente trabajo, la visibilidad entre dos nodos depende en gran medida de la presencia de edificios obstructores (Figura 4). Las ecuaciones que modelan las pérdidas de propagación han sido extraídas del modelo urbano micro-celular desarrollado en el proyecto europeo WINNER [15].

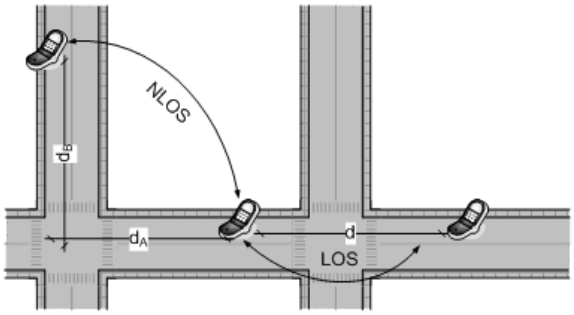


Figura 4. Condiciones de propagación en entornos urbanos

El tráfico modelado para medir las prestaciones de los algoritmos se muestra esquematizado en la Figura 5, y sigue los patrones de tráfico a ráfagas actualmente experimentados en la transmisión de datos. Es importante destacar que el modelo empleado no pretende reproducir de forma exacta un cierto tipo de tráfico, sino modelar las intermitencias características en las transmisiones de datos y sesiones de Internet. En el modelo implementado, durante sesiones de 200 segundos se lleva a cabo un tráfico intermitente compuesto por periodos de actividad (*ON*) de duración 5 segundos y periodos de inactividad (*OFF*) de 15 segundos. Durante los periodos de *ON* son enviados en dirección a la BS un total de 50 paquetes, es decir, 10 paquetes/segundo. El periodo de inactividad de 15 segundos provoca la destrucción de las tablas de rutas de los nodos por expiración de la validez y por tanto, la necesidad de una nueva búsqueda de ruta en el siguiente periodo de *ON*.

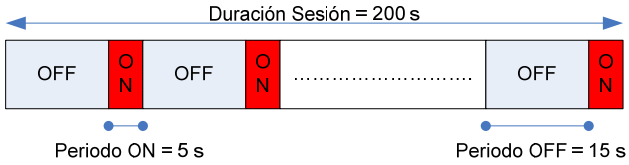


Figura 5. Modelo de tráfico ON-OFF

TABLA 1

PARÁMETROS DE CONFIGURACIÓN PARA LA SIMULACIÓN

Parámetro	Valor
Numero de nodos	500
Potencia de Transmisión (W)	0.2
Tipo de escenario	Manhattan o cuadrícula
Distribución de los nodos no uniforme	Doble vía en avenidas centrales
Dimensiones del escenario (m x m)	2250 x 2250
Anchura de los edificios (m)	225
Anchura de calle (m)	25
Velocidad (Distribución normal (media ,var)) (m/s)	N(1.5, 0.1^2)
Modelo de movilidad	Random Walk Obstacle
Tecnología de acceso radio	802.11a
Tasa de transmisión (Mbps)	12
Protocolo de enrutamiento	AODV modificado
Periodo de Beacon (s)	5
Tiempo de simulación (s)	10000
Numero de nodos fuente	50
Duración de la sesión de tráfico (s)	200
Duración periodo ON (s)	5
Tasa de envío en ON (pkt/seg)	10
Tamaño de paquete (bytes)	500
Duración periodo OFF (s)	15
Modelo de propagación	Pathloss LOS NLOS

V. RENDIMIENTO DE PROTOCOLOS DE ENRUTAMIENTO MULTI-HOP EN REDES MCN-MR

Con el fin de analizar el efecto que la consideración de la direccionalidad tiene en el funcionamiento y rendimiento de los protocolos de enrutamiento multi-hop en las redes MCN-MR, la presente sección estudia en primer lugar la técnica Multi Métricas (MM) definida en [12]. Esta técnica ha sido escogida como algoritmo con el que comparar las propuestas del presente artículo debido a su alto rendimiento en redes inalámbricas multi-hop. A continuación, se presentarán y analizarán las distintas propuestas del presente artículo que incorporan la direccionalidad con el fin de optimizar el rendimiento de los protocolos de enrutamiento reduciendo su señalización y consumo energético.

A. Multi Métricas (MM)

La función de costes definida en [12] y presentada en la sección III considera el número de saltos, la congestión del canal y la energía consumida para identificar la ruta óptima entre origen y destino. En el presente artículo, el rendimiento de MM es evaluado considerando la modificación de AODV establecida por el estándar 802.11s, combinación referida en [12] como MMRP-I pero que por sencillez va a referirse en el presente artículo como MM.

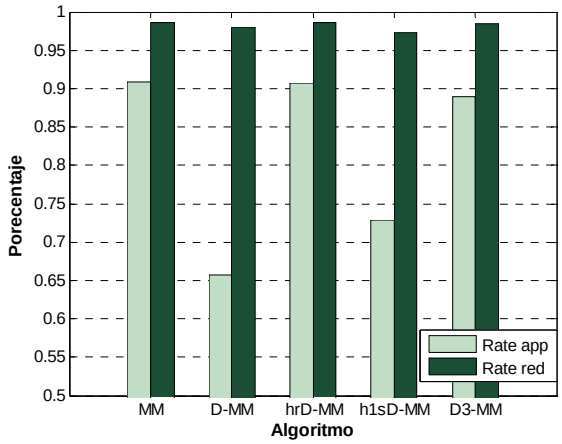


Figura 6. Tasa de paquetes entregados al destino respecto a los generados por la aplicación y a los enrutados.

Como puede apreciarse en la Figura 6, la técnica MM consigue altas tasas de paquetes entregados al destino con respecto a los generados por la aplicación (rate_app), y de paquetes entregados al destino en relación al número de paquetes enrutados (rate_red). El alto rendimiento de MM es debido a su función de costes y al protocolo de enrutamiento emulado, el cual realiza búsqueda de rutas en todas las direcciones, siendo finalmente escogida la ruta de mínimo coste. Si bien, este procedimiento de búsqueda de rutas resulta en una alta eficiencia en el enrutamiento de paquetes en redes inalámbricas multi-hop, también genera una elevada señalización (Figura 7) que puede resultar en una alta congestión del canal radio. Esta congestión viene reflejada por el número de paquetes de búsqueda de ruta que han sido retransmitidos por nodos intermedios en cada intento de búsqueda de ruta (*RREQ* retransmitidos / *RREQ*). El hecho de que la búsqueda de las rutas se realice en todas direcciones y que los nodos intermedios sean capaces de reenviar un mismo RREQ si posee un coste acumulativo menor era necesario para garantizar un alto rendimiento, pero es también el causante de la elevada congestión.

B. Direccionalidad-Multi Métricas

Con el fin de reducir la señalización y congestión resultante del protocolo de enrutamiento establecido por el estándar 802.11s junto a la función de costes MM, los autores del presente artículo proponen la consideración de la direccionalidad en el proceso de enrutamiento a través del protocolo Direccionalidad-Multi Métricas (D-MM), el cual explota el conocimiento de la ubicación de la BS en su proceso de búsqueda de ruta. A diferencia del protocolo MM, D-MM no permite que nodos intermedios que estén más alejados del destino que lo estaba el nodo anterior retransmitan los mensajes de búsqueda de ruta; de este modo, todos los saltos suponen una progresión hacia la BS. En el proceso de búsqueda de ruta, el paquete *RREQ* es descartado si, tras ejecutarse el algoritmo que calcula la directividad hacia el destino, se comprueba que el nodo que recibe el *RREQ* está más alejado del nodo destino *D* que lo estaba el nodo que envió el *RREQ*. De entre las rutas que vayan dirigidas hacia *D*, D-MM escoge aquella que minimiza la función de costes definida por MM.

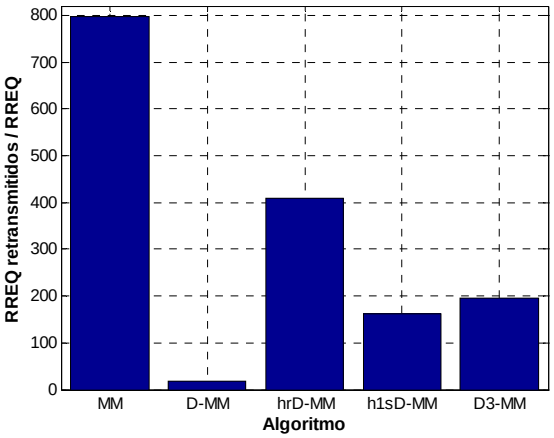


Figura 7. Numero de paquetes de RREQ reenviado por los nodos intermedios por intento de búsqueda de ruta

TABLA 2

NÚMERO DE SALTOS Y DISTANCIA RECORRIDA EN LAS RUTAS CREADAS

Algoritmo	MM	D-MM	hrD-MM	h1sD-MM	D3-MM
Saltos	7.157	5.632	7.109	6.349	7.075
Distancia de ruta (m)	1191.9	980.79	1192.8	1075.1	1178.2

Como puede observase en la Figura 7, el protocolo D-MM consigue reducir de forma muy notable la señalización y congestión generada por el protocolo de enrutamiento multi-hop; la reducción de la señalización comparada con el algoritmo MM es del 97%. Sin embargo, la 'radicalidad' en la consideración de la direccionalidad por el protocolo D-MM resulta en una importante reducción de la tasa de paquetes entregados al destino. Dicha reducción es debida a la imposibilidad de creación de ruta por parte de D-MM; el 95% de los paquetes que no han sido entregados al destino no han conseguido salir del nodo origen *S*, es decir, no ha sido posible establecer una ruta y el propio nodo *S* los ha descartado. Por el contrario, una vez establecida una ruta, el protocolo D-MM obtiene una alta tasa de paquetes entregados con respecto a los enrutados. En este caso, el protocolo D-MM incrementa con respecto a MM (18%) el throughput medido a nivel de aplicación (Figura 8). Dicha diferencia se explica por el hecho de que la consideración de la direccionalidad en D-MM reduce el número de saltos y la distancia recorrida entre origen y destino (Tabla 2) con respecto al protocolo MM. Con el fin de reducir las desventajas que caracterizan el

rendimiento de D-MM, los autores proponen otros protocolos que intentan mantener el rendimiento de MM reduciendo su coste de señalización al nivel de D-MM al incorporar también la direccionalidad en el proceso de establecimiento de ruta.

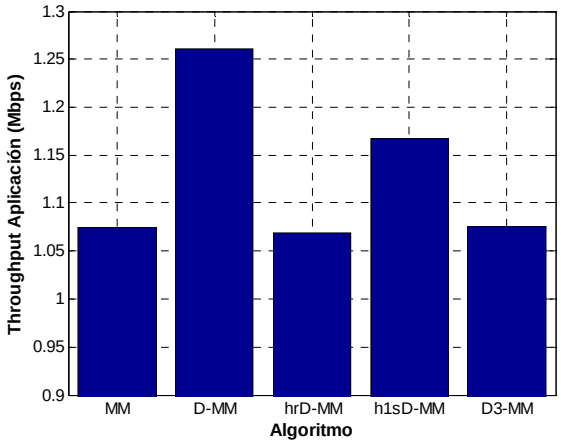


Figura 8. Throughput a nivel de aplicación

C. Híbrido ruta Direccionalidad-Multi Métricas

La propuesta de protocolo híbrido ruta Directividad-Multi Métricas (hrD-MM) es un punto intermedio entre las versiones MM y D-MM en el proceso de búsqueda de ruta. Aprovechando que el algoritmo de enrutamiento realiza varios intentos de búsqueda de ruta si transcurrido un tiempo no recibe respuesta (*RREP*) del nodo destino *D*, el algoritmo hrD-MM realiza un primer intento de búsqueda de ruta con la versión D-MM, y si no consigue establecer una ruta los sucesivos intentos son realizados con la versión MM para intentar maximizar la probabilidad de establecimiento de ruta entre los nodos fuente y destino. Cuando el nodo *S* inicia el proceso de búsqueda de ruta, un temporizador limita el tiempo permitido al paquete *RREQ* para encontrar una respuesta del nodo *D*. Durante este periodo permitido de búsqueda, ningún otro paquete *RREQ* es enviado por la estación *S*. Si transcurrido este periodo el nodo *S* no ha recibido el paquete *RREP* de *D*, el protocolo de enrutamiento entiende que el proceso de búsqueda de ruta ha sido fallido, e inicia de nuevo el proceso con un nuevo *RREQ* y el procedimiento establecido por MM. En consecuencia, hrD-MM es una solución híbrida entre D-MM y MM que intenta en primer lugar ofrecer el rendimiento de D-MM, y si no lo consigue opera según el protocolo MM. La Figura 6 muestra que hrD-MM cumple con su primer objetivo en cuanta a tasa de paquetes entregados, aunque de nuevo reduciendo el throughput de aplicación (Figura 8). Por el contrario, hrD-MM, si bien no consigue reducir los niveles de señalización al nivel de D-MM, sí que consigue una reducción muy notable (48%) con respecto al protocolo MM. Como puede observarse en la Tabla 3, el protocolo hrD-MM también consigue un menor porcentaje de rutas que se rompen del total de rutas creadas, y una mayor duración de las rutas que se rompen con respecto a MM.

TABLA 3

PORCENTAJE DE RUTAS ROTAS DEL TOTAL DE RUTAS CREADAS Y DURACIÓN DE LAS RUTAS

Algoritmo	MM	D-MM	hrD-MM	h1sD-MM	D3-MM
Rutas rotas (%)	32.4	25.5	28.8	31.8	31.2
Duración (s)	1.581	1.804	1.718	1.696	1.6841

D. Híbrido 1-salto Direccionalidad-Multi Métricas

La siguiente propuesta de los autores, el protocolo híbrido 1-salto Direccionalidad-Multi Métricas (h1sD-MM), busca un nuevo compromiso, reducir el nivel de congestión e incrementar el throughput de aplicación conseguidos por el algoritmo hrD-MM. Para ello, h1sD-MM acota la región de búsqueda de ruta, pues únicamente aquellos nodos intermedios que están situados a una distancia inferior a la que se encuentra el nodo *S* de la BS tienen permiso para retransmitir el *RREQ*. Para ello, el nodo *S* introduce en el paquete *RREQ* la distancia a la que se encuentra de BS cuando inicia el proceso de búsqueda de ruta. Los nodos intermedios comprueban este campo al recibir el mensaje *RREQ*, y tan solo si su distancia es inferior a la que marca el paquete colaboran en la búsqueda de la ruta. Este funcionamiento se corresponde con la utilización de D-MM para el primer salto del mensaje *RREQ*, y una versión acotada de MM para el resto de reenvíos.

Las Figuras 7 y 8 muestran que los objetivos perseguidos por h1sD-MM en relación a la congestión y al throughput de aplicación han sido conseguidos. h1sD-MM reduce en un 80% la congestión causada por MM, y en un 60 % la causada por hrD-MM. El throughput de aplicación ha conseguido incrementarse en un 8.5% respecto a MM y hrD-MM, aunque sigue siendo inferior a D-MM, lo que resulta en una reducción de la tasa de paquetes entregados (Figura 6). Los resultados mostrados en la Tabla 2 reflejan como h1sD-MM, debido a las restricciones que impone en el proceso de búsqueda de ruta, consigue construir las rutas con un menor número de saltos, consiguiendo pues reducir la distancia recorrida entre los nodos origen y destino.

E. Direccionalidad x-permisos Multi Métricas

Por último, los autores proponen un último protocolo Direccionalidad x-permisos Multi Métricas (Dx-MM) que trata de fructificar las sinergias de los algoritmos vistos anteriormente, considerando el compromiso existente entre los objetivos buscados. Dx-MM es una versión más 'permisiva' del algoritmo D-MM al considerar la direccionalidad en el proceso de búsqueda de ruta entre los nodos fuente y destino. Para ello, el paquete *RREQ* contiene un campo en el que se indica el número de saltos permitidos (x-permisos) en dirección contraria al nodo destino *D*. Con el protocolo Dx-MM, el paquete *RREQ* se propagará por la red a medida que las rutas vayan dirigidas hacia el nodo *D*, o no hayan sido sobrepasados el número de permisos establecidos por el nodo *S*. Los resultados mostrados para el protocolo D3-MM en las figuras 6 y 8 muestran como D3-MM consigue una tasa de paquetes entregados similar al protocolo MM con un throughput de aplicación ligeramente superior. Además, el

protocolo D3-MM consigue reducir de forma muy notable, con respecto a MM, el nivel de señalización y congestión generados por el proceso de establecimiento de ruta multi-hop (Figura 7). Los resultados obtenidos han demostrado que la variación en el número de permisos ofrece un compromiso entre señalización/congestión y éxito en la tasa de entrega de paquetes al destino.

Por último, y con el fin de enfatizar el beneficio de considerar la direccionalidad en el proceso de enrutamiento multi-hop para redes MCN-MR, conviene analizar la energía consumida por parte de los nodos móviles al implementar los distintos protocolos analizados. Como ya fue destacado en la introducción, el consumo energético es un factor clave en la operatividad de las redes MCN-MR. La Figura 9 muestra el consumo energético generado por la señalización de la creación de rutas (denominada *Routing*) en comparación con el total de la energía consumida. Los resultados obtenidos muestran que en el caso del protocolo MM, la señalización necesaria para el proceso de enrutamiento supone un 55% del consumo total energético de los nodos. La figura 9 también muestra como las distintas propuestas presentadas en este artículo permiten reducir de forma significativa el consumo energético en el proceso de establecimiento de ruta, siendo interesante destacar al protocolo D3-MM, el cual a pesar de alcanzar el rendimiento MM consigue reducir notablemente el consumo energético. La Tabla 4 muestra la energía media consumida por los nodos y la homogeneidad en el consumo energético de los distintos nodos emulados (desviación típica). La tabla 4 muestra de nuevo que los algoritmos propuestos en este artículo realizan un consumo energético más eficiente que el algoritmo MM. Lo más destacable de la Tabla 4 es que la consideración de la direccionalidad en el proceso de enrutamiento no sólo puede reducir el consumo energético sino también hacerlo más homogéneo entre los nodos móviles.

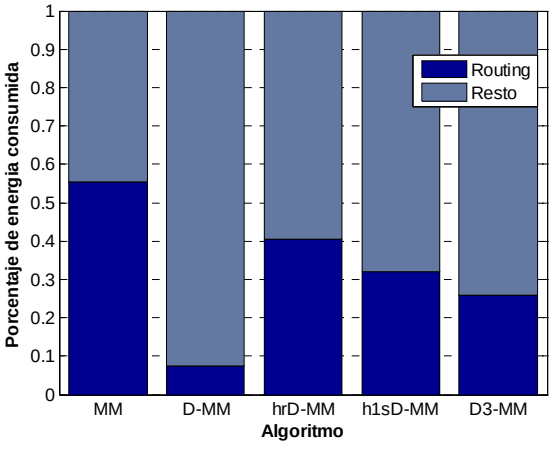


Figura 9. Porcenataje de energía consumida

TABLA 4

ENERGÍA MEDIA CONSUMIDA Y DESVIACIÓN TÍPICA

Algoritmo	MM	D-MM	hrD-MM	h1sD-MM	D3-MM
Energía consumida (J)	0.334	0.1456	0.2581	0.2051	0.2111
Desviación típica	0.0626	0.0474	0.06	0.0625	0.0753

VI. CONCLUSIÓN

En el presente artículo, los autores han propuesto una serie de algoritmos de enrutamiento, basados en la modificación de AODV establecida en el estándar 802.11s y la función de costes MM definida en [12], que aprovechan las características de las redes MCN-MR para conseguir altos rendimientos en el proceso de enrutamiento, reduciendo a su vez la señalización y consumo energético resultantes del proceso de enrutamiento. En concreto, los protocolos propuestos abogan por explotar la direccionalidad en las redes MCN-MR al poder conocer la ubicación del nodo destino en los enlaces inalámbricos multi-hop. Los resultados obtenidos han demostrado que los protocolos propuestos permiten establecer rutas fiables y eficientes que consiguen reducir notablemente el consumo energético.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el Ministerio de Ciencia e Innovación bajo el proyecto TEC2008-06728 y por el Ministerio de Industria, Turismo y Comercio bajo el proyecto TSI-020400-2008-113.

REFERENCIAS

[1] International Telecommunication Union. [Online] .Sitio Web Oficial. http://www.itu.int.

[2] Y. Lin y Y. Hsu, "Multihop Cellular: A New Architecture for Wireless Communications", *en IEEE Proceeding INFOCOM,* 2000.

[3] R. Ananthapadmanabha, B.S. Manoj y C.S.R. Murthy, "Multi-hop Cellular Networks: The Architecture and Routing Protocols", *en IEEE Symposium PIMRC,* 2001.

[4] C. Perkins y E. Royer, "Ad hoc On-Demand Distance Vector Routing", *en IEEE Proceedings Workshop on Mobile Computing Systems and Applications*, 1999.

[5] IEEE P802.11s/D2.0, draft amendment to standard IEEE 802.11: Mesh Networking. *IEEE Standard,* 2007.

[6] C. E. Perkins y P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", *en ACM Proceeding SIGCOMM'94: Computer Communications Review*, 1994.

[7] S. Lee y D. Cho, "On-Demand Energy-Efficient Routing for Delay-Constrained Service in Power-Controlled Multihop Cellular Network", *en IEEE Proceedings VTC2004-Fall,* 2004.

[8] T. Aure y F. Y. Li, "Load-balancing in Multi-homed OLSR Networks using Airtime Metric and Radio-aware Path Selection: Implementation and Testing" white paper, Universidad de Agder, 2008,

[9] L. T. Nquyen y R. Beuran, "A load Aware Routing Metric for Wireless Mesh Networks", *en IEEE Symposium on Computers and Communications*, 2008.

[10] T. Salonidis, M. Garetto, A. Saha y E. Knightly , "Identifying High Throughput Paths in 802.11 Mesh Networks: a Model-based Approach", *en IEEE International Conference ICNT*, 2007.

[11] Y. Park y E. Jung, "Resource-Aware Routing Algorithms for Multi-hop Cellular Networks", *en IEEE International Conference MUE,* 2007.

[12] L. Cao, K. Sharif, Y. Wang y T. Dahlberg, "Adaptive Multiple Metrics Routing Protocols for Heterogeneous Multi-Hop Wireless Network", *en IEEE Proceedings CCNC*, 2008

[13] The Network Simulator – ns2. Sitio web oficial: http://www.isi.edu/nsnam/ns/

[14] K. Maeda, A. Uchiyama, T. Umedu, H. Yamaguchi, K. Yasumoto y T. Higashino, "Urban Pedestrian Mobility for Mobile Wireless Network Simulation", *en Elsevier Ad Hoc Networks*, 2009.

[15] WINNER, "D1.1.1. WINNER II interim channel models", *Public Deliverable*, https://www.ist-winner.org/index.html

Descripción semántica de aplicaciones web mediante microservicios

José Ignacio Fernández-Villamor, Carlos Ángel Iglesias, Mercedes Garijo
Departamento de Ingeniería de Sistemas Telemáticos,
Universidad Politécnica de Madrid
Avenida Complutense nº 30, "Ciudad Universitaria". 28040 - Madrid (España)
jifv@dit.upm.es, cif@dit.upm.es, mga@dit.upm.es

***Resumen*—Las posibilidades de desarrollo de aplicaciones en la web actual están limitadas por la interoperabilidad entre aplicaciones y servicios. La investigación en el campo de la web semántica, servicios web semánticos y mashups busca contribuir a la interoperabilidad y composición de servicios y aplicaciones, aunque muchas alternativas no han logrado suficiente adopción debido a su complejidad.**

En este artículo se propone un método de clasificación de servicios ligero para el estilo arquitectónico REST, presentándose la idea de Microservicios. Frente a anteriores enfoques de servicios web semánticos que buscan la descripción de cualquier tipo de servicio web, en el método propuesto se pretende poder definir descripciones subóptimas sencillas y naturales, llamadas descripciones de microservicios. Las descripciones de microservicios consisten en una lista de términos que representan características de un servicio. Los beneficios de desarrollar una aplicación web mediante microservicios son la generación automática de documentación de la interfaz de programación, la posibilidad de automatizar las pruebas de la aplicación, o la capacidad de descubrir, ejecutar y componer servicios de aplicaciones web.

***Palabras Clave*—REST, aplicaciones web, web programable, servicios web**

I. Introducción

La web actual posee un creciente número de aplicaciones y servicios que cubren diferentes dominios de interés. Actualmente, los usuarios disfrutan de un amplio abanico de aplicaciones, desde aplicaciones de comercio electrónico a blogs, noticias o redes sociales. Las posibilidades de la web actual están limitadas únicamente por la interoperabilidad entre aplicaciones. El número de casos de uso de Internet se multiplicaría si las aplicaciones pudiesen componerse arbitrariamente y ser ejecutadas automáticamente para satisfacer la meta de un usuario. Este enfoque de construir una web programable ha llevado a la investigación en múltiples áreas, como servicios web semánticos y mashups o combinaciones de aplicaciones. Muchos de estos enfoques han sido exitosos desde un punto de vista investigador, pero no han sido capaces de lograr gran adopción debido a su alta complejidad y a su falta de integración con la arquitectura de Internet.

En este artículo se propone un método ligero de anotación y clasificación de servicios que sigue el estilo arquitectónico REST y que posibilita descubrimiento, ejecución, composición, documentación y pruebas automáticas de aplicaciones web. Dichas descripciones se realizan en un lenguaje natural, buscándose la facilidad de llegar a descripciones subóptimas sencillas de servicios, sin pretender cubrir todo el espacio de servicios existente en Internet.

En el artículo, en primer lugar se describe el estado del arte de tecnologías similares y estudios relacionados, así como la problemática existente. En segundo lugar, proponemos nuestro método de clasificación analizando sus características y casos de uso. Finalmente, se resumen las principales conclusiones del trabajo de investigación realizado.

II. Trabajos relacionados

Hay diversos enfoques que contribuyen a la mejora de la integración de información y la composición de servicios.

La arquitectura de servicios web [4] y los estándares de servicios web semánticos que se definen sobre ella (como OWL-S [14], WSMO [11], METEOR-S [10] o WSDL-S [13]) proporcionan un enfoque pesado al descubrimiento, ejecución y composición de servicios. Algunas alternativas han sido empleadas por empresas en sus buses de servicios, mientras que las alternativas semánticas han sido principalmente utilizadas en contextos de investigación. En general, puede decirse que esta arquitectura no ha gozado de gran adopción en Internet. Su falta de integración con el estilo arquitectónico REST y la alta complejidad de los estándares pueden ser razones de este fracaso.

Para mejorar la integración con la arquitectura de Internet, se comenzaron a utilizar enfoques REST en aplicaciones web, implementándose interfaces de programación separadas para las funcionalidades más comunes de las aplicaciones. Para mejorar la automatización y la interoperabilidad, se han propuesto múltiples alternativas con un enfoque REST.

SA-REST [12] propone anotar las especificaciones de interfaces de programación (APIs) para permitir descubrimiento, ejecución y composición automática de servicios REST. El enfoque asume que existe una API para funcionalidades comunes de la aplicación junto con una especificación textual disponible públicamente. Tabmién asume que la parte de las aplicaciones que son consumidas por humanos no son procesadas por agentes automáticos. Sobre ello, SA-REST propone anotar las especificaciones textuales de APIs con RDFa [2] para habilitar el procesado automático de APIs. Los esfuerzos de desarrollo siguiendo el enfoque de SA-REST residen en construir una API separada para las funcionalidades más comunes de la aplicación, escribir una especificación textual de la API y anotar esta especificación con RDFa para habilitar el procesado automático. hRESTs [15] es un enfoque similar, que propone un conjunto de microformatos para permitir la anotación de especificaciones de interfaces de programación.

Finalmente, el lenguaje de descripción de aplicaciones web (WADL) [7] es otro enfoque REST para descripción

de interfaces de servicios que, en una forma similar a SA-REST, propone definir un fichero XML que describe una API REST. El fichero WADL describe la API de una forma autocontenida y, a diferencia de SA-REST, no reutiliza la información disponible en la especificación textual de la API.

III. Descripción del problema

A. Los principios de diseño de Internet

La web fue creada con el objetivo de construir un sistema extensible y distribuido de hipermedios siguiendo el estilo arquitectónico REST (*Representational State Transfer*). REST consiste en un conjunto de restricciones que pueden ser aplicadas sobre cualquier otro sistema distribuido de hipermedios. En el caso de Internet, limita la arquitectura a un sistema cliente-servidor cacheable, en capas, sin estado, con facilidades de código bajo demanda y con una interfaz uniforme para sus recursos. Estas prácticas se utilizan en el diseño de la web, y han sido ignoradas ocasionalmente, llevando a desajustes en protocolos y problemas en aplicaciones web [5].

Respecto a los aspectos sociales relacionados con el desarrollo web, los desarrolladores normalmente desean reducir al máximo la complejidad de las aplicaciones. Típicamente, los enfoques en integración de la información y descripción de servicios están basados en gran medida en descripciones formales de servicios. Especificar formalmente el modelo de interfaz y proceso de servicios y APIs es una tarea que suele evitarse por su complejidad, a no ser que proporcione un valor añadido importante a la aplicación. Como conclusión, las iniciativas para hacer la web programable (por ejemplo, los mashups o los servicios web semánticos) necesitan equilibrar los esfuerzos de formalización con los beneficios a nivel de interoperabilidad y automatización.

Por ello, es importante considerar estos aspectos para definir una solución para automatización de servicios que se adapte a la filosofía de Internet y tenga adopción entre desarrolladores.

B. Limitaciones de la interfaz uniforme

Como se ha dicho anteriormente, la web actual sigue el estilo arquitectónico REST. El protocolo HTTP proporciona una forma de acceder a los recursos mediante una interfaz común que está basada, principalmente, en cuatro métodos o verbos: GET, PUT, DELETE y POST. Esto implica que la manipulación de un recurso ha de realizarse utilizando exclusivamente uno de estos verbos.

La seguridad y la idempotencia son propiedades que poseen algunos de estos métodos. Las operaciones seguras son aquellas que no tienen ningún tipo de efecto secundario sobre el estado de la web. Las operaciones idempotentes son aquellas cuyo resultado es el mismo tanto si han sido ejecutadas una vez como si han sido ejecutadas más veces. Una operación segura es, por tanto, idempotente.

El método HTTP GET solicita una representación de un recurso indicado por su identificador o URI, PUT almacena una representación de un recurso proporcionada como parámetro en una determinada URI, DELETE elimina un recurso especificado por su URI y POST procesa una representación dada como subordinada de un recurso dado por su URI. Por ello, GET es una operación segura e idempotente, PUT y DELETE son operaciones idempotentes pero inseguras y POST es una operación insegura y no idempotente.

¿Cómo afecta el estilo arquitectónico REST al desarrollo de aplicaciones web? Las aplicaciones web tienen un dominio o campo de interés (como, por ejemplo, el comercio electrónico o las redes sociales) en el que se involucran ciertas operaciones de negocio y conceptos particulares. Siguiendo el estilo arquitectónico REST, los conceptos deberían modelarse como recursos REST, mientras que las operaciones de negocio deberían modelarse como métodos de la interfaz uniforme.

Siguiendo la semántica de los métodos HTTP, las operaciones de lectura deberían asociarse con operaciones GET, las operaciones de almacenamiento como PUT, las de eliminación como DELETE y cualquier otra operación como POST. Como resultado, no es posible conocer la semántica de una operación que se realiza tras un método, al existir un infinito rango de operaciones de negocio realizable que se asocia a sólo cuatro métodos posibles.

Por ejemplo, en el caso de una aplicación de comercio electrónico, una operación de vaciado del carro de la compra debería implementarse mediante el método POST. Por otro lado, hacerse amigo de otra persona en una red social es otra operación que debería implementarse con un método POST. Ambas operaciones utilizan el mismo método de la interfaz uniforme de REST, pero no tienen el mismo significado ni el mismo efecto.

Por ello, se necesita información adicional para que agentes autónomos sepan la operación precisa que un método HTTP realiza, sin importar si esta información puede ser inferida del contexto o si se especifica mediante anotaciones. Esta es una desventaja de utilizar una interfaz uniforme en el estilo arquitectónico REST.

IV. Microservicios

En este artículo proponemos un enfoque para la descripción de servicios en aplicaciones web que sigue el estilo arquitectónico REST y que pretende tener un equilibrio entre efuerzos de desarrollo y el valor añadido proporcionado.

Otros enfoques proponen construir servicios paralelos a la parte de las aplicaciones web que es utilizada por los usuarios humanos. Para reducir esfuerzos de desarrollo, proponemos anotar las aplicaciones web de forma que los usuarios humanos y los agentes automáticos puedan seguir el mismo flujo de trabajo al utilizar una aplicación, en la línea de iniciativas como RDFa y POSH [8].

La filosofía que hay detrás de este enfoque reside en proporcionar a los clientes automáticos los mismos medios que los usuarios humanos para utilizar las aplicaciones web. Al utilizar una aplicación web, los usuarios humanos interaccionan con ella en respuesta a las salidas que proporciona la aplicación, siguiendo enlaces y enviando formularios de acuerdo a un comportamiento esperado de la aplicación que se deduce del contenido de las páginas web, tanto a nivel textual como contextual.

Iniciativas como los microformatos contribuyen a proporcionar una forma de describir los contenidos de las representaciones de los recursos web. En este enfoque, sin embargo, falta una manera de describir las interacciones con los recursos más allá de la semántica del método HTTP utilizado. Considerando un servicio en una aplicación como una combinación de un

recurso y un método HTTP, se propone a continuación la idea de microservicios para resolver esta problemática y así posibilitar la utilización de aplicaciones web por parte de agentes automáticos.

La implementación de una aplicación con microservicios permite varias funcionalidades adicionales, tales como:

- Descubrimiento de servicios automático por parte de motores de búsqueda.
- Descubrimiento, ejecución y composición automática de servicios por agentes inteligentes y combinadores de aplicaciones o *mashups*.
- Generación automática de documentación de servicios.
- Prueba automática de servicios.
- Asistencia en la navegación web, como rellenado de formularios.

A. Introducción a los microservicios

La web actual posee multitud de aplicaciones con gran cantidad de servicios diferentes a disposición de los usuarios. Existen múltiples iniciativas para describir servicios, como OWL-S o WSMO. Estas iniciativas intentan ofrecer un método para describir cualquier servicio posible, obteniéndose como resultado estándares muy complejos de utilizar. En este artículo proponemos un enfoque diferente, en el que se proporciona un conjunto cerrado de elementos para describir servicios, intentando hacer posible definir una descripción subóptima para cualquier tipo de servicio. Dado que está inspirado en la idea de Microformatos, llamamos Microservicios [6] a estos servicios subóptimos que pueden describirse. En otras palabras, suponiendo un espacio multidimensional en el que existen los diferentes servicios, los microservicios pretenden ser servicios con máxima similaridad con servicios de un mismo tipo.

Siguiendo el enfoque de descripción de servicios de WSMO, la semántica de un servicio se define a través de un conjunto de precondiciones, asunciones, postcondiciones y efectos [11] que, en general, son un conjunto de condiciones o restricciones requeridas y que se aplican ante la ejecución satisfactoria del servicio. Estas condiciones describen, por ejemplo, efectos secundarios, presencia de parámetros de entrada, o la semántica de la salida del servicio.

Las restricciones de un servicio pueden agruparse de acuerdo a las características de alto nivel que pretenden describir. Por ejemplo, si se necesita como parámetro una lista de palabras clave en un servicio de búsqueda, este hecho puede describirse con una precondición que exija la presencia de una variable de entrada `keywords` y una postcondición que implique que la salida debe incluir las palabras clave proporcionadas en `keywords`. Ambas condiciones buscan añadir una característica de filtrado por palabras clave al servicio, dando lugar a un elemento descriptivo reutilizable para otros servicios. El conjunto de características disponibles da lugar a la "caja de herramientas" que los desarrolladores web pueden utilizar para describir un microservicio.

En las siguientes subsecciones se definirá el concepto de microservicios en detalle mediante un modelo y se presentarán los intermediarios de microservicios.

B. Modelo de descripción de microservicios

Las descripciones de microservicios son descripciones ligeras para aplicaciones web siguiendo el estilo arquitectónico REST. En esta sección, el concepto de microservicio se clarifica definiendo un modelo de descripción de microservicios.

De aquí en adelante consideraremos un servicio como una interacción con un recurso de una aplicación web, es decir:

Definición 1. Un *servicio* es una dupla (r, m) que tiene un recurso r y un método m.

Por tanto, un recurso puede tener tantos servicios asociados como métodos HTTP existen. Siguiendo esta definición de servicio, se define el concepto de microservicio.

Definición 2. Un *microservicio* es un servicio que está descrito por una descripción de un microservicio y por su descripción extendida asociada.

Los conceptos de descripción de microservicio y descripción extendida se definen a continuación.

Definición 3. Una *descripción de microservicio* es un conjunto de términos t_1, t_2, ..., t_n. Cada término t_i representa una característica que tiene el servicio.

La descripción de un microservicio puede ser transformada en una descripción extendida mediante una función de transformación. Una descripción extendida de un servicio se define siguiendo el enfoque WSMO de descripción de servicios.

Definición 4. Una *descripción extendida de servicio* s es un conjunto de precondiciones, postcondiciones y descripciones textuales que describen un servicio. Es decir, $s \in \mathcal{P}(\mathcal{PRE} \cup \mathcal{POS} \cup \mathcal{TEX})$, siendo $\mathcal{PRE}$, $\mathcal{POS}$, y $\mathcal{TEX}$ los conjuntos de precondiciones, postcondiciones y descripciones textuales, respectivamente, y $\mathcal{P}(X)$ el conjunto potencia de X.

La función de transformación utiliza definiciones de términos para construir la descripción extendida.

Definición 5. Una *definición de un término* d es un conjunto de precondiciones, postcondiciones y descripciones textuales que se asocian a un conjunto de términos. Es decir, $d \in \mathcal{P}(\mathcal{T}) \times \mathcal{P}(\mathcal{PRE} \cup \mathcal{POS} \cup \mathcal{TEX})$, siendo $\mathcal{T}$ el conjunto de términos y $\mathcal{P}(\mathcal{T})$ su conjunto potencia.

Las precondiciones, postcondiciones y descripciones textuales en las definiciones cuyos términos están presentes en la descripción del microservicio se utilizan por la función de transformación para producir la descripción extendida del servicio.

Definición 6. Una *función de transformación* F es una función que combina definiciones de términos en una descripción de un microservicio para producir una descripción extendida. Es decir, $F : \mathcal{M} \rightarrow \mathcal{S}$, con $F(m) = \bigcup_i C_i \mid (T_i, C_i) \in \mathcal{D} \wedge T_i \subseteq m$, siendo $\mathcal{D}$, $\mathcal{M}$ y $\mathcal{S}$ el conjunto de definiciones, el conjunto de descripciones de servicios y el conjunto de descripciones extendidas de servicios, respectivamente.

Por tanto, la función de transformación construye una descripción extendida de un servicio de acuerdo con un vocabulario de términos. Las capacidades descriptivas de las

descripciones de servicios imponen los límites sobre qué es un microservicio y qué no lo es. Por ello, dado un determinado vocabulario de términos, algunos servicios serán microservicios al poder ser descritos mediante descripciones de microservicios, y otros no.

Un ejemplo de descripción de microservicio puede ser `keyword-filtered multiple picture get`, que consta de cuatro términos, y describe un servicio de búsqueda de imágenes por palabras clave. Su descripción extendida se muestra en la figura 1. Puede observarse que la descripción extendida posee una descripción textual, precondiciones y postcondiciones, posibilitando documentación automática, pruebas automáticas o ejecución automática, entre otras funcionalidades, como se detallará en la sección V.

Dicha descripción extendida se obtiene, de acuerdo con la definición 6, agrupando precondiciones, postcondiciones y descripciones textuales de aquellas definiciones de términos cuyos términos aparecen en la descripción del servicio. La descripción del ejemplo se construye mediante la combinación de las definiciones `keyword-filtered`, `multiple get`, `picture get` y `get`.

Este ejemplo ilustra la necesidad de que las condiciones impuestas por un término puedan ser dependientes de otros términos presentes en la descripción. Por ejemplo, el término `picture` impone una postcondición ("devolver una imagen") al aplicarse sobre `get`, pero impone una precondición ("una imagen debe proporcionarse como parámetro") al aplicarse sobre `post`.

En resumen, los microservicios proporcionan una descripción dual de servicios: (i) una descripción estándar, que consiste en una lista de términos que representan características de alto nivel y (ii) una descripción extendida, que proporciona una especificación formal del servicio. Ambas descripciones son procesables por agentes automáticos o humanos, mientras que utilizar una u otra depende del nivel de expresividad necesario para la tarea requerida.

C. Descripciones en lenguaje natural

El enfoque utilizado para las descripciones de microservicios encaja de forma natural con el lenguaje hablado. Cada término de una descripción añade restricciones y condiciones de la misma forma a cómo se hace en lenguaje natural. Por ello, eligiendo nombres para los términos a un alto nivel de abstracción permite obtener una descripción prácticamente en lenguaje natural.

Los métodos HTTP están representados por sus correspondientes términos. `get`, `post`, `put` y `delete` son términos que implican la utilización del respectivo método HTTP en el microservicio, añadiendo las convenientes precondiciones y postcondiciones.

Para incrementar la semántica de los métodos HTTP, pueden añadirse más términos para construir una descripción con más significado. Por ejemplo, el microservicio `picture post` implica que la entidad enviada es una imagen, mientras que `multiple get` implica que el recurso solicitado es una lista de recursos en un formato de lista acordado, como XOXO [9]. Una variante de este último ejemplo sería el microservicio `keyword-filtered multiple picture get`, que describiría un servicio similar a Flickr en el que se devuelven múltiples imágenes filtradas por unas palabras clave. Finalmente, `picture private sharing post` describiría un servicio en el que una imagen es enviada para su compartición con un conjunto privado de usuarios de una determinada red social.

Puede observarse que todas las descripciones son amigables y naturales, lo que hace que su construcción sea fácil y sencilla dado un determinado vocabulario de términos con el que trabajar.

D. Microservicios intermediarios

Con esta filosofía, rara vez habrá un microservicio que encaje perfectamente con otro servicio. En los casos en que no haya encaje, o bien el servicio se debe adaptar a la descripción del microservicio, o se ha de definir un intermediario que utilice el servicio pero se ajuste a la descripción del microservicio. En esta sección se tratará este último caso, la utilización de intermediarios o *proxies* de microservicios.

Definición 7. Un *microservicio intermediario* es un microservicio que (i) ejecuta otro servicio y (ii) tiene una descripción de microservicio que es la descripción subóptima del servicio externo.

Al utilizar un microservicio intermediario, el cliente del servicio debe conocer el servicio que se ejecuta realmente. Por ejemplo, http://www.flickr.com/search permite buscar imágenes en el dominio http://www.flickr.com. Si se implementa un microservicio intermediario con ese servicio y se despliega en http://ejemplo.com/flickr/search, el cliente podría asumir erróneamente que el intermediario busca imágenes en el dominio http://ejemplo.com. Por ello, la descripción del microservicio para el intermediario ha de ser complementada con el servicio que realmente se ejecuta para permitir una correcta interpretación del funcionamiento del intermediario.

Definición 8. Una *descripción de microservicio intermediario* es una descripción de microservicio seguida de una URI al servicio que el intermediario ejecuta.

Por tanto, el microservicio intermediario del ejemplo anterior podría describirse como `keyword-filtered multiple picture get http://www.flickr.com/search`. Informalmente, podría entenderse como "http://ejemplo.com/flickr/search permite ejecutar http://www.flickr.com/search como un microservicio `keyword-filtered multiple picture get`".

En resumen, los microservicios intermediarios permiten reutilizar servicios y utilizarlos como microservicios sin modificar la implementación original. Esto permite una transición suave hacia una aplicación basada totalmente en microservicios en un sistema ya desplegado.

V. Utilización de microservicios

La utilización de microservicios en una aplicación web y su descripción permiten una serie de funcionalidades añadidas que se resumen en esta sección.

A. Generación de documentación

La documentación acerca de la utilización de un microservicio se genera automáticamente a partir de la descripción del microservicio. Dicha documentación es la descripción extendida del microservicio, la cual se produce mediante la

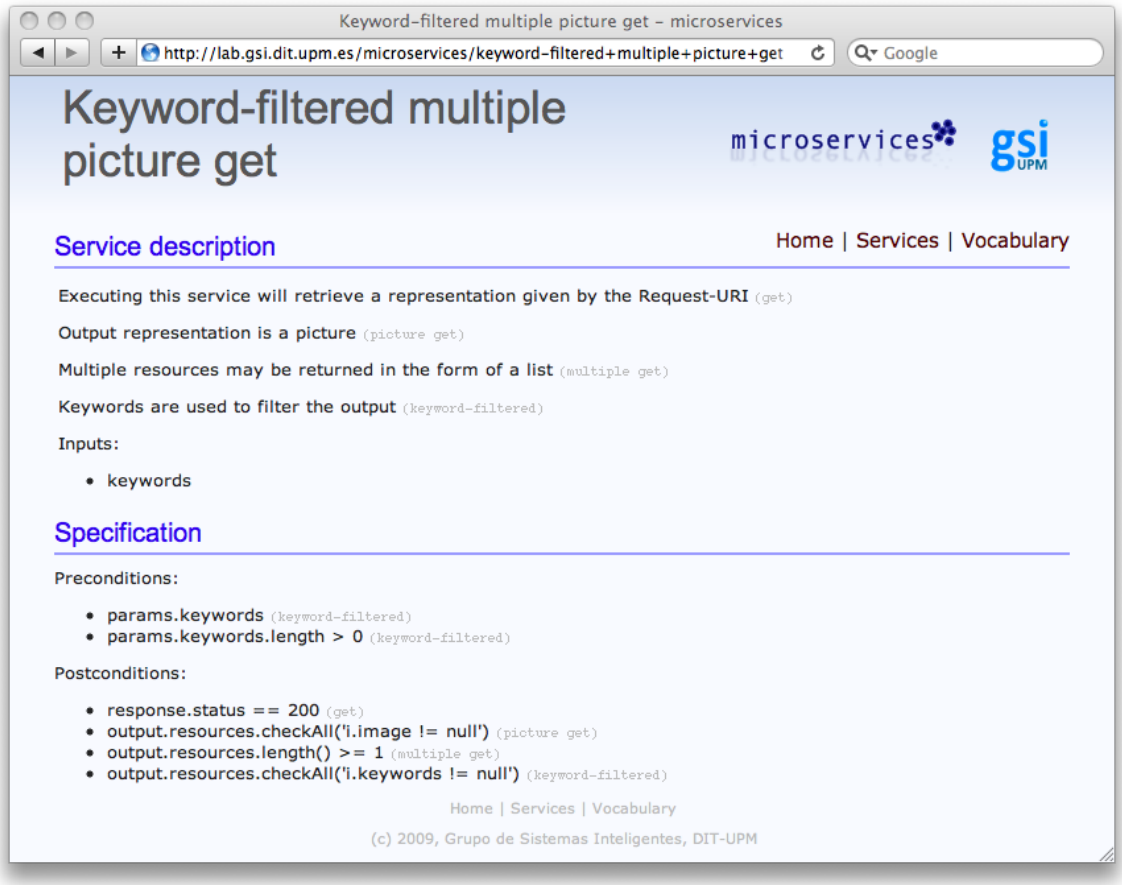


Fig. 1. Ejemplo de descripción extendida de microservicio de búsqueda de imágenes por palabras clave

función de transformación. Dicha descripción consta de dos partes:

- Descripción textual: La descripción textual se obtiene mediante la concatenación de las descripciones de cada uno de los términos.
- Descripción formal: Son las precondiciones y postcondiciones del servicio, que se obtienen mediante la unión de las condiciones que aporta cada término de la descripción. Se expresan mediante un lenguaje de acuerdo a un determinado modelo de sistema, que contiene elementos como petición, respuesta, etc.

La descripción textual es comprensible por un agente humano, mientras que la descripción formal es apta para agentes automáticos y también para humanos en caso de que se desee clarificar detalles del funcionamiento del servicio.

B. Descubrimiento automático

Considerando una aplicación web cuyas interacciones con los recursos son microservicios, pueden incluirse anotaciones en las representaciones de los recursos de la aplicación para que un agente automático pueda conocer las interacciones que se llevan a cabo en dicha aplicación.

Un esquema posible para permitir descubrimiento automático sería incluir la descripción del microservicio en un parámetro de la etiqueta de formulario HTML, como en el siguiente ejemplo en el que se enriquece un formulario de un microservicio de búsqueda:

```
<form
 class="keyword-filtered multiple get"
 action="/search"
 method="get">
    <input name=keywords value="">
    <input name=btnG type=submit value="Search">
</form>
```

Esta modificación permite que un agente rastreador que recorra la web siguiendo hipervínculos pueda conocer la semántica del formulario. De esta manera, utilizando la descripciones de los microservicios un agente automático puede construir una base de datos de microservicios disponibles en la web y realizar filtrados por términos para buscar, por ejemplo, microservicios de búsqueda (`multiple get`), de imágenes (`picture`), etc.

Otra opción más pesada consiste en utilizar la descripción extendida, de forma análoga a la utilización de servicios web semánticos WSMO. Al diseñar un agente inteligente que descubra microservicios a partir de su descripción extendida, se le debería proporcionar la capacidad de procesar las precondiciones y postcondiciones de la descripción para inferir qué microservicios satisfacen sus metas determinadas. Ésta sería la opción más compleja al tener que utilizarse el modelo de sistema empleado en las condiciones de la descripción extendida, en lugar de los términos de la descripción normal.

Consultar la presencia de ciertos términos en la descripción de un microservicio simplifica en gran medida una consulta de encaje de servicios frente al enfoque más formal de procesar las precondiciones y postcondiciones del servicio. En función

de la tarea requerida, puede ser más apropiado utilizar una u otra alternativa.

C. Ejecución automática

Si las interacciones posibles en una aplicación web son microservicios, esto permite ejecutar la aplicación por parte de un agente automático dada la información proporcionada de la semántica de dichas interacciones. A este respecto, al igual que en el caso del descubrimiento, existen dos niveles de semántica utilizables, por la dualidad existente en las descripciones de microservicios.

Nuevamente, la opción más pesada consiste en utilizar la descripción extendida de los microservicios. En este caso, un agente inteligente capaz de ejecutar un microservicio a partir de su descripción extendida procesaría las condiciones de la descripción para ejecutar el servicio de acuerdo a los detalles de formato o parámetros que se muestran en las condiciones.

Utilizando únicamente la descripción normal basada en lista de términos, la implementación de un agente que consuma un microservicio debería conocer de antemano el acuerdo de implementación de servicio que conlleva cada término de la descripción.

Así, por ejemplo, si un término `keyword-filtered` implica que las palabras clave deben estar separadas por espacios, éste hecho podría: (i) incluirse de forma directa en la implementación del agente, de acuerdo al enfoque más ligero que sólo considera las descripciones normales o (ii) inferirse de las postcondiciones en el caso de utilizar la descripción extendida.

La implementación de un consumidor genérico de servicios que infiera cada hecho de las condiciones existentes en una descripción es una tarea compleja que puede no aportar el suficiente valor añadido en muchos casos. Por ello, en general la utilización de la descripción extendida para inferir formatos, parámetros o efectos para la ejecución de un microservicio no debería ser necesaria.

D. Pruebas automáticas

La descripción extendida de un microservicio posibilita la validación de casos de prueba de forma automática. Para:

- un servicio s con precondiciones $pre(s)$ y postcondiciones $pos(s)$,
- un conjunto de elementos del sistema (como parámetros o recursos web) $e_1...e_n$,
- dos estados del sistema s_{bef} (antes de la ejecución del servicio) y s_{aft} (después de la ejecución),

se obtiene que:

$$pre(e_1(s_{bef})...e_n(s_{bef})) \Rightarrow pos(e_1(s_{aft})...e_n(s_{aft})) \quad (1)$$

Por otro lado, se define un caso de prueba como sigue:

Definición 9. Un *caso de prueba* es un conjunto de elementos del sistema en un estado determinado $e_1(s_1)..e_i(s_1)$.

Y, teniendo en cuenta la expresión 1, se verifica que, para un caso de prueba T:

$$\forall e \in T, pre(e(s_{bef})) \Rightarrow pos(e(s_{aft})) \quad (2)$$

Es decir, dado un conjunto de casos de prueba, con la descripción extendida de un microservicio se puede determinar el subconjunto de casos de prueba que ejecutan el microservicio de forma correcta, gracias a las precondiciones. Además, se dispone de los criterios de validación de cada caso de prueba, gracias a las postcondiciones.

Así, la prueba de servicios se reduce a la definición de un conjunto de casos de prueba, entendidos como un simple conjunto de parámetros de entrada y de recursos web, automatizándose la verificación de la validez de dichos casos de prueba y la definición de las condiciones de validación para la salida.

E. Extensibilidad

Como se ha dicho, la idea de microservicios es simplificar las descripciones de servicios. Para ello, el vocabulario de términos con el que trabajar debe mantenerse reducido por simplicidad y para favorecer la reutilización, pero tan amplio como sea posible para permitir la descripción de muchos servicios. A continuación se enumeran algunos principios para mantener un buen vocabulario de términos para su utilización en descripciones de microservicios:

- Un término debe representar una característica de alto nivel de abstracción. Es discutible cuándo una característica es de alto nivel o no. Sin embargo, una posible regla puede ser que un término debería tener el mismo nivel de abstracción que el vocabulario empleado en las aplicaciones web, en las que se manejan términos como blogs, compartición, productos, amigos, etc.
- El nombre de un término debería ser fácilmente comprensible por un humano, y encajar de la mejor forma posible en una frase en lenguaje natural.
- Aparte de los métodos HTTP, un término no debería representar una acción. Los métodos HTTP son el único conjunto de acciones permitidas en la web por la interfaz uniforme del estilo arquitectónico REST. Estas acciones pueden ser extendidas mediante términos que representen roles, como `private`, `keyword-filtered` o `sharing`.
- Un término no debería representar al mismo tiempo un rol y un tipo de datos. Por ejemplo, podrían definirse dos términos como `comment-upload` y `comment-param` en lugar de simplemente `comment`. En el primer caso se amplía gratuitamente el vocabulario, mientras que en el segundo caso se aplica una dependencia con los términos `get` y `post` para definir un único término, reduciendo el vocabulario y simplificando el descubrimiento de microservicios.

VI. Trabajos relacionados

Existen algunos trabajos de investigación que se centran en la descripción de formularios HTML. RDF-Forms [3] busca "incorporar a la Web Semántica capacidades similares a los formularios HTML". Así, propone esquemas que representan a cada uno de los métodos HTTP, anotando sus entradas semánticamente, de forma opuesta al enfoque de microservicios, en el que las entradas quedan anotadas por las postcondiciones del servicio.

OpenSearch [1] es una alternativa para el modelado de servicios de búsqueda para compartir resultados de búsqueda. Al contrario que los microservicios, OpenSearch está enfocado a un tipo muy concreto de servicio, lo que permite especificar con un alto nivel de detalle este tipo de servicios de búsqueda.

Su filosofía es también diferente, dado que en OpenSearch existe un descriptor separado, definido para cada servicio de búsqueda, en el que se especifican características como el tipo de resultados, idioma o codificaciones.

VII. Conclusiones

En este artículo se han revisado enfoques para la automatización de servicios en la web actual. Por regla general, dichas alternativas han sufrido carencias como baja integración con REST, el estilo arquitectónico de Internet, o bien alta complejidad, lo que ha reducido su adopción generalizada.

Como alternativa, se ha propuesto el concepto de microservicios en aplicaciones web para cubrir de forma subóptima la tarea de descripción de servicios. Como se ha visto, la descripción de microservicios está al nivel de sencillez del lenguaje natural, al tiempo que permiten la automatización de tareas como documentación, pruebas, descubrimiento y ejecución, beneficiando a campos como los agentes inteligentes o la combinación de aplicaciones.

Como trabajos futuros se plantean tareas en la línea de facilitar la adopción de este esquema de descripción semántica en aplicaciones web ya existentes. Un primer paso es definir reglas para la clasificación automática de servicios en sus microservicios subóptimos. Para ello, habría que analizar conjuntos de servicios y ver cómo se manifiestan las diversas características que presentan. Por ejemplo, la existencia de viñetas o listas podrían ser indicadores de un formato de salida de lista de recursos, representado por el término `multiple`. Análogamente, un servicio cuya salida contiene las palabras clave incluidas como entrada es susceptible de ser filtrado por palabras clave y contener el término, etc. El resultado sería un conjunto de reglas heurísticas de términos para clasificación automática de servicios en sus microservicios más semejantes. Esta clasificación automática sería complementable con la generación automática de intermediarios.

Agradecimientos

Este trabajo de investigación ha sido financiado por la Comisión Europea bajo el proyecto de I+D ROMULUS (FP7-ICT-2007-1) y por el Gobierno Español bajo el proyecto de I+D Java sobre Ruedas (FIT-350401-2007-8).

Referencias

[1] A9.com, inc. OpenSearch specification. http://www.opensearch.org/Specifications/OpenSearch/1.1, 2005.
[2] B. Adida and M. Birbeck. RDFa Primer - Bridging the Human and Data Webs. http://www.w3.org/TR/xhtml-rdfa-primer/, 2008.
[3] M. Baker. RDF Forms. http://www.markbaker.ca/2003/05/RDF-Forms/, 2005.
[4] W. Consortium. Web Services Architecture. http://www.w3.org/TR/ws-arch/, 2004.
[5] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, 2000.
[6] Grupo de Sistemas Inteligentes. Microservices. http://lab.gsi.dit.upm.es/microservices, 2009.
[7] M. J. Hadley. Web application description language. https://wadl.dev.java.net/wadl20061109.pdf, 2006.
[8] Microformats community. Plain old semantic html (posh). http://microformats.org/wiki/posh, 2007.
[9] Microformats community. Microformats. http://microformats.org/, 2008.
[10] A. Patil, S. Oundhakar, A. Sheth, and K. Verma. METEOR-S Web service Annotation Framework. In *Proceeding of the World Wide Web Conference*, 2004.
[11] D. Roman, U. Keller, H. Lausen, J. de Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler, and D. Fensel. Web Service Modeling Ontology, Applied Ontology. IOS Press.
[12] A. P. Sheth, K. Gomadam, and J. Lathem. SA-REST: Semantically Interoperable and Easier-to-Use Services and Mashups. In *IEEE Computer Society*, 2007.
[13] World Wide Web Consortium. Web Service Semantics - WSDL-S. http://www.w3.org/Submission/WSDL-S/.
[14] World Wide Web Consortium. OWL-S: Semantic Markup for Web Services. http://www.w3.org/Submission/OWL-S/, 2004.
[15] Wright State University. HTML Microformat for Describing RESTful Web Services and APIs. http://knoesis.wright.edu/research/srl/projects/hRESTs/#hRESTs, 2008.

EXPERIENCIAS Y PERSPECTIVAS DE ENTORNOS DE APRENDIZAJE 3D COLABORATIVOS

M. B. Ibáñez, J. J. García Rueda, S. Galán, D. Maroto, C. Delgado Kloos.
Departamento de Ingeniería Telemática,
Universidad Carlos III de Madrid
Av. Universidad, 30, Edif. Torres Quevedo. E-28911 Leganés (Madrid).
{mbibanez,rueda}@it.uc3m.es, {sgalan,dmaroto}@inv.it.uc3m.es, cdk@it.uc3m.es

***Resumen*- Los entornos 3D son la última interfaz llegada al mundo de las relaciones persona-ordenador, y sobre todo de las relaciones persona-persona mediadas por ordenador, y ya desde el primer momento está empezando a emplearse con fines educativos. Desde actividades constructivas colaborativas en SecondLife, pasando por reuniones virtuales en Wonderland o conexiones inmediatas entre mundos en Croquet, los experimentos educativos en las principales plataformas 3D del momento se multiplican. Como base para un futuro desarrollo más sólido de estas experiencias, en el presente artículo se hará un repaso de lo que estos entornos ofrecen a día de hoy, tras previamente haber analizado lo que sería conveniente que ofreciesen de cara a un aprovechamiento más completo de su potencial educativo.**

***Palabras Clave*- colaboración, 3D, didáctica, plataformas.**

I. INTRODUCCIÓN

En cualquier espacio en el que se reúne un grupo de personas aparecen formas de colaboración. Estas formas pueden ir desde la más completa espontaneidad hasta las actividades de colaboración perfectamente planificadas. En el caso de los mundos virtuales 3D, también es posible encontrar ambas formas de colaboración, pero aunque si bien las dos se están empleando a día de hoy con fines educativos, es la segunda, la colaboración planificada, la que augura mejores rendimientos educativos, así como un avance más significativo en los próximos años.

Para hacer un resumen casi testimonial de lo que supone a día de hoy la colaboración espontánea, antes de entrar con la que más nos interesa aquí, diremos que es éste un tipo de colaboración que se da fundamentalmente en los juegos online multijugador, también denominados MMORGs ("*Massively Multiplayer Online Role-playing Game*"), tales como los conocidos "*World of Warcraft*" y "*Everquest*", por ejemplo [1]. Estos juegos son, probablemente, uno de los exponentes económicamente más importantes del mundo 3D actual, y desde luego una fuente de colaboración de lo más interesante. Diariamente miles de jugadores se conectan a los servidores de estos juegos y se embarcan en campañas conjuntas que implican el empleo y desarrollo de estrategias de colaboración ciertamente sofisticadas:

- Colaboración espontánea entre grupos numerosos.
- Elaboración instantánea de estrategias.
- Formación de grupos estables o puntuales.
- Colaboración/cooperación con objetivos bien definidos.
- Surgimiento de roles.
- Entrenamiento en coordinación de equipos.

Este tipo de actividades y comportamientos, claramente deseables en un contexto educativo, pueden surgir y emplearse también en la colaboración planificada, pero entonces su surgimiento debe diseñarse, controlarse, fomentarse y mantenerse, de manera que el proceso colaborativo produzca los resultados de aprendizaje esperados. Tradicionalmente, son las técnicas de diseño instruccional las que proporcionan las herramientas conceptuales necesarias para esta correcta planificación de las actividades didácticas y su integración con el resto de la experiencia de aprendizaje. Desde el *boom* de la tecnología educativa a mediados de los 90, estas técnicas se han ido aproximando paulatinamente a las Tecnologías de la Información y las Comunicaciones, creando nuevas técnicas a aplicar en estos entornos pero también beneficiándose de estas tecnologías para mejorar y facilitar los procesos del propio diseño instruccional.

Ahora estas técnicas y metodologías didácticas han de dar el salto a los sistemas 3D, encontrando formas efectivas de aprovechar las posibilidades que estos nuevos entornos ofrecen a fin de conseguir aplicarlos con éxito a los procesos de enseñanza y aprendizaje. Para ello existen dos posibles aproximaciones.

La primera de ellas es observar las características de las plataformas 3D actuales e idear formas de aprovecharlas en beneficio del aprendizaje. Esta aproximación, claramente orientada al corto plazo, permite una aplicación inmediata de los elementos ya existentes, lo cual es claramente una ventaja deseable. Sin embargo, en contrapartida parece ahondar en la tendencia de que la didáctica vaya siempre un paso por detrás de la tecnología, cuando dado el enorme potencial de estos nuevos entornos quizá sería muy conveniente empezar permitiendo a la didáctica plantear sus necesidades y expectativas. Esta segunda aproximación es la que adoptaremos en este trabajo, en el que partiremos de los requisitos que debería ofrecer un entorno 3D colaborativo orientado al aprendizaje para posteriormente, y bajo ese prisma, analizar las características que ofrecen algunas de las plataformas 3D más populares en la actualidad, prestando especial atención a las características de dichas plataformas que permiten satisfacer los requisitos previamente planteados.

Así las cosas, este artículo está organizado de la siguiente manera. La sección II presenta el conjunto de requisitos de los entornos de aprendizaje colaborativo organizados en los tres ejes que nos ocupan: el virtual, el colaborativo y el de aprendizaje. La sección III contempla los requisitos que deben satisfacer las plataformas donde se implementarán los entornos educativos. En la sección IV se enumeran los servicios que brindan tres plataformas de inmersión virtual (Second Life, Croquet y Wonderland) y en la sección V se describen algunas aplicaciones educativas realizadas sobre esas plataformas. Por último, en la sección VI se analiza la factibilidad de implantación de entornos educativos colaborativos virtuales dada la madurez tecnológica actual y se esboza el entorno educativo que nos proponemos realizar.

II. REQUISITOS DE LOS ENTORNOS DE APRENDIZAJE COLABORATIVOS

Para favorecer la colaboración en ambientes virtuales educativos, es necesario no sólo que el usuario pueda sentirse inmerso en la experiencia educativa sino que además tenga mecanismos que articulen su trabajo. Todo esto sin olvidar que el usuario debe ser capaz de manejar adecuadamente la tecnología a su disposición.

La inmersión en el ambiente virtual se alcanza gracias a la representación de sí mismo que hace el usuario por medio de avatares, la calidad de la escenografía del mundo virtual y la posibilidad de interacción que el usuario tiene con el entorno virtual mediante interfaces que extienden sus sentidos.

Los aspectos recién señalados han sido desarrollados con relativo éxito en el mundo de los juegos virtuales. Sin embargo, los ambientes educativos colaborativos virtuales requieren además, una serie de servicios de colaboración provistos de servicios de contenido en donde los estudiantes puedan usar la tecnología para comunicarse con quien desean en el tiempo y espacio que sea mejor para ellos siguiendo el proceso educativo pactado.

Para permitir los servicios de colaboración es necesario sentar las bases para permitir y fomentar [2]:

- Tareas de trabajo: Como por ejemplo planificación, tormenta de ideas y creatividad grupal, toma de decisiones, negociación, competición, diseminación de información.
- Tareas de transición: Se refiere a las tareas empleadas para pasar de una tarea de trabajo a otra tanto en colaboración síncrona como asíncrona. Ejemplos: reasumir tareas previas, asignar trabajo, preparar la reunión virtual.
- Protocolos sociales: Definen la manera en que la sesión se lleva a cabo, permitiendo definir el grado de formalidad o informalidad, la dinámica de la sesión y lo que se espera de cada uno de los presentes.
- Formación de grupo: Se articula en torno a tres dimensiones: tiempo, tipo de grupo y relativas al sistema computador. Entre las actividades incluidas en la dimensión tiempo se encuentran la duración de la sesión, la duración del esfuerzo colaborativo completo, y si la sesión es espontánea o está planeada. La dimensión del tipo de grupo tiene en cuenta el número de componentes, desde cuándo ese grupo lleva trabajando junto, grado de homogeneidad entre sus miembros, distribución geográfica y necesidades de sincronía o asincronía. La dimensión relativa al sistema computador se ocupa de la plataforma *hardware* con que cuenta el grupo, el tiempo disponible para formar a los participantes y del nivel de conocimientos informáticos de los mismos.

Los servicios de colaboración antes mencionados deben apoyar la comunicación y permitir compartir información. En entornos 2D se utilizan para estos propósitos herramientas del tipo Web 2.0, el mundo 3D está aún por explorar.

Aparte de la comunicación gestual que puede ser realizada mediante los avatares, la comunicación requerida puede ser vía texto, audio y vídeo y debe poder llevarse a cabo tanto de manera síncrona como asíncrona. De modo pues que en estos entornos debe ser factible llevar a cabo la comunicación en todas esas dimensiones.

Es necesario además, poder compartir información de manera implícita y explícita. La colaboración explícita se lleva a cabo compartiendo al menos, documentos y objetos, mientras que la colaboración implícita, más sutil y menos explorada tecnológicamente, se espera que ayude a la toma de decisiones en base, por ejemplo, a comportamientos registrados de (o por) otros miembros de la comunidad social. En este sentido, están ganando popularidad tanto la llamada folcsonomía como la exhibición de nubes de etiquetas.

Por último, en cuanto a los servicios académicos se requiere gestionar evaluaciones, orquestar el desarrollo del curso, y brindar servicios académicos en general. Estos servicios son los más específicos del área educativa, al menos desde su vertiente más administrativa: el entorno debe ofrecer al profesor herramientas que le permitan llevar a cabo las tareas habituales de gestión de una acción docente, como por ejemplo editar, distribuir y corregir pruebas de evaluación tanto sumativas como formativas, desarrollar la planificación del curso y vincular sus distintas actividades al resto de servicios ofrecidos por el entorno, ofrecer la posibilidad de publicar calificaciones, avisos, calendarios de actividad, etc. En definitiva, los entonos de aprendizaje deberían permitir la realización de las tareas tradicionalmente asociadas a la planificación didáctica.

III. REQUISITOS DE LAS PLATAFORMAS

Entendemos por plataforma, un entorno de *software* donde una aplicación puede ser desarrollada, probada y ejecutada. En el caso que nos ocupa, en la plataforma debe poder desplegarse un entorno educativo virtual 3D que dé soporte tecnológico a las necesidades establecidas en el apartado anterior.

Como primer paso para el establecimiento de un entorno virtual colaborativo se espera que la plataforma permita orquestar un sistema distribuido en el que los usuarios puedan entrar y salir independientemente de su ubicación física. Dos tipos de sistemas son comunes en estos casos, la arquitectura cliente-servidor o la arquitectura *Peer To Peer* (P2P). En el primer caso, el usuario se conectará al sistema a través del servidor central y en el segundo caso a través de cualquier otro nodo del sistema distribuido. Los nodos del sistema deben intercambiar información entre sí de forma que en tiempo real todos los usuarios puedan ver el mundo virtual desde su perspectiva.

Para lograr la inmersión en el ambiente virtual, la plataforma debe brindar la posibilidad no solo de configurar

físicamente el avatar que representa al usuario sino además, establecer las bases para la comunicación de forma verbal o no verbal entre los avatares presentes en la escena. La comunicación verbal requiere que la plataforma maneje audio en tiempo real. Para la comunicación no verbal se debe contar al menos con *chat* de texto, y con la posibilidad de dotar al avatar con una amplia gama de gestos. Además, una cámara debe permitir ver el entorno virtual desde la perspectiva del avatar, esto implica no solo que el avatar tenga libertad de movimiento sino también que la plataforma tenga un motor de renderizado 3D. La inmersión en la escena puede ser aún mayor si la plataforma permite acoplar dispositivos hápticos para una mejor manipulación u observación de los objetos de la escena.

Para satisfacer las necesidades de comunicación básicas, es necesario permitir el intercambio de texto síncrono, que en la Web 2.0 se realiza en tiempo real gracias a la Mensajería Instantánea (IM), y el intercambio asíncrono: correo electrónico (*e-mail*) y foros. Con este propósito, sobre las plataformas debe ser posible implantar *chat,* transferencia de archivos, listas de contactos y establecer conversaciones simultáneas.

Las comunicaciones técnicamente más complejas, las de audio y vídeo, resultan imprescindibles en plataformas que soportan ambientes virtuales. La comunicación síncrona por audio permite establecer conversaciones uno-a-uno, uno-a-muchos. Existen también formas de comunicación por audio asíncronas donde es factible grabar y reproducir conversaciones. En portales de emisoras de radio se utilizan ambos tipos de comunicaciones. La video conferencia provee la comunicación por vídeo síncrona por excelencia. Sin embargo, las herramientas de vídeo masivamente utilizadas son las asíncronas entre las que vale la pena mencionar You Tube [3].

En cuanto al requerimiento de los entornos educativos ligado a compartir información existen herramientas que permiten realizar esta tarea en ambientes colaborativos 2D. En estos entornos, son de consumo masivo los sitios web (*websites)*, las *wikis*, los *blogs*, la suscripción a información vía RSS, los sitios para compartir imágenes, fotos (por ejemplo Flickr [4]), los sitios para compartir documentos (por ejemplo Docs de Google), los repositorios de documentos y las herramientas para el control de versiones, por mencionar solo los más representativos. Se espera que en los ambientes virtuales educativos 3D sea posible tener variaciones de estas herramientas adaptadas al nuevo medio.

Para fomentar las tareas de trabajo de la aplicación, la plataforma debe tener no solo mecanismos para la importación, creación, borrado y composición de objetos a partir de otros más simples, sino también un lenguaje de guiones que permita que los objetos reaccionen ante estímulos. Cabe esperar que los objetos de los entornos educativos sean relevantes al proceso de aprendizaje particular.

La herramienta educativa debe proporcionar capacidades relacionadas con las tareas de transición necesarias no sólo en mundos colaborativos sino principalmente en los ligados al aprendizaje. Las capacidades ligadas a este tipo de tareas son la localización de colaboradores, establecimiento de agenda de trabajo, manejo de calendarios, posibilidad de hacer sondeos y de repetir experiencias. Para ello la plataforma debe ofrecer la posibilidad de programar este tipo de transiciones.

Las capacidades ligadas a los protocolos sociales involucran por ejemplo, indicadores de presencia, control de turno de palabra, sincronización, comunicación en grupo, espacios de trabajo privados. Las posibilidades de manejo del sistema distribuido junto con los elementos de *chat* de texto y de voz de la plataforma, cubren estas capacidades.

Las capacidades relacionadas con la formación de grupos tienen que ver con actividades *ad-hoc* para este propósito. En el campo tecnológico, las capacidades están ligadas a diferentes formas de comunicación entre los participantes.

Por último, los servicios académicos a nivel administrativo estarán cubiertos siempre y cuando la plataforma permita trabajar de forma distribuida, provea de un lenguaje de programación lo suficientemente rico como para soportar aplicaciones similares a las requeridas por un ambiente Web 2.0. Aquellos servicios más ligados al campo pedagógico necesitarán de todo el soporte que la plataforma pueda brindar en cuanto a entornos virtuales colaborativos.

IV. PLATAFORMAS DE INMERSIÓN VIRTUAL

Las plataformas que se analizan en este apartado han sido escogidas por la entidad sin fines de lucro conocida como *Immersive Education Initiative* [5] para establecer un ecosistema de plataformas en el que puedan intercambiarse objetos de aprendizaje. En la *Immersive Education Initiative* colaboran universidades, institutos de investigación, consorcios y compañías a nivel internacional uniendo sus esfuerzos para definir y desarrollar estándares abiertos, plataformas y comunidades capaces de promover sistemas de aprendizaje basados en mundos virtuales.

A. Second Life

Second Life (SL) [6] es un entorno virtual 3D al que se accede vía Internet y que es creado de manera colaborativa por sus usuarios (residentes). Esta plataforma ha sido desarrollada por Linden Research, Inc. [7] con fines comerciales.

Second Life trabaja sobre Windows, Linux y Mac OS, requiere de al menos 256MB de memoria RAM (512MB en Mac OS). Se recomienda una CPU que trabaje al menos a 800MHz (1GHz para Mac OS) y una tarjeta gráfica similar a nVidia GeForce2.

En el ambiente inmersivo provisto por Second Life, los usuarios pueden crear avatares completamente configurables capaces de realizar una amplia gama de movimientos. Existe además la posibilidad de crear, modificar objetos, e intercambiar diversidad de productos virtuales a través de un mercado abierto que tiene como moneda local el Linden Dólar ($L). La comunicación entre los avatares también es posible y se lleva a cabo por texto, voz y correo electrónico.

Cualquier aspecto del mundo puede ser programado, y el comportamiento de los objetos puede ser controlado mediante el uso de eventos, todo esto gracias al lenguaje de guiones *Linden Scripting Language* (LSL) creado específicamente para SL.

El modelo de negocio desarrollado por Linden Lab en torno a esta plataforma, establece el uso limitado del mundo virtual. Las herramientas y el lenguaje de guiones son de libre

acceso pero, para poseer tierra y poder construir en ella, es necesario crear una cuenta de pago.

Second Life utiliza una arquitectura cliente-servidor. El servidor central maneja clusters de máquinas que permiten tanto la simulación del mundo virtual como el almacenamiento de los datos, un pequeño grupo de máquinas se encargan de autenticar a los usuarios. Para minimizar el tiempo de comunicaciones el mundo está dividido en 256 x 256 regiones, los objetos de una región son responsabilidad de un único servidor.

La plataforma Second Life, es estable, tanto el usuario como el desarrollador tienen a su disposición suficiente documentación.

B. Croquet

Croquet [8], [9] es una plataforma de *software* libre que permite la creación de ambientes de colaboración virtuales distribuidos en tiempo real. El proyecto Croquet persigue tanto el desarrollo como el despliegue de simulaciones colaborativas, entornos experimentales y laboratorios virtuales para la industria, la investigación y la educación. Croquet es el fruto de la investigación llevada a cabo por sus arquitectos, Alan Kay, David A. Smith. David P. Reed, Andreas Raab, Julian Lombardi, y Mark McCahill.

Croquet trabaja sobre Windows, MacOS y Linux. Requiere de un computador con una antigüedad no mayor de 2 años, una tarjeta gráfica igual o mejor que nVidia GeForce2 y sus requisitos de ancho de banda son similares a los de un IP Voice Conferencing System.

Croquet es una extensión del lenguaje de programación orientado-objeto Squeak [10], [11] que define un sistema distribuido P2P. Croquet se ha beneficiado de diversos desarrollos realizados sobre Squeak, a saber: Morphic [12] que permite la creación interactiva de interfaces usuario, eToys [13] para la programación de entidades virtuales y TeaTime [14] para compartir y establecer comunicación entre objetos distribuidos. En la actualidad se está desarrollando el proyecto Cobalt [15] para dotar a Croquet con un navegador de espacios virtuales.

Las facilidades de construcción y manipulación de objetos se lleva a cabo gracias a Morphic. Morphic trabaja con objetos gráficos llamados *Morphs* que pueden ser escogidos, movidos, incluidos dentro de otros objetos, redimensionados, rotados y borrados.

eToys [13] aporta las facilidades de *scripting* que permiten a los objetos exhibir comportamientos como respuesta a eventos. eToys es un ambiente de programación ampliamente utilizado en educación para niños y es una referencia obligada en entornos colaborativos educativos 2D.

La comunicación y la sincronización de objetos se realizan gracias a la arquitectura TeaTime [14] diseñada para soportar un número masivo de usuarios interactuando concurrentemente en un espacio virtual. TeaTime trabaja mediante la replicación y sincronización de los elementos del sistema, toda modificación sobre un objeto es también realizada sobre sus réplicas mediante la propagación de mensajes.

La plataforma emplea la OpenGL para las representaciones gráficas y OpenAL para el manejo del audio. Por ahora los avatares son muy rudimentarios, los esfuerzos del proyecto han estado centrados en el desarrollo del espacio y portales. En Croquet, un espacio es un lugar 3D con objetos con los que se puede interactuar, mientras que un portal es la interfaz 3D que facilita al usuario la transportación a los espacios del resto de los usuarios.

Croquet se encuentra en etapa de desarrollo pre-alpha y por ahora dispone de soporte técnico escaso. En cuanto a las facilidades para el desarrollador tenemos que la máquina virtual de Squeak así como la de Croquet está escrita en Squeak y permite la modificación del código en tiempo de ejecución.

C. Wonderland

Wonderland [16] es un proyecto de código libre patrocinado por Sun Microsystems Labs [17]. El objetivo original del proyecto fué permitir el desarrollo de entornos colaborativos, distribuidos, escalables y robustos en los que se pudieran realizar actividades de negocio. En la actualidad, se está promocionando como una plataforma en donde también pueden desarrollarse entornos educativos.

Wonderland es una arquitectura distribuida cliente-servidor basada en JAVA [18], trabaja bajo los Sistemas Operativos Linux, Solaris, MacOS y Windows XP, requiere que los clientes estén sobre PC de al menos 1GB RAM, 1.5Ghz y tengan acelerador gráfico. Wonderland integra otros cuatro proyectos de desarrollo e investigación de Sun Microsystems Labs también de código libre y desarrollados en JAVA: Proyecto Darkstar [19], jVoiceBridge [20], jMonkeyEngine [21] y Proyecto Looking Glass [22].

El Proyecto Darkstar [19] provee la infraestructura escalable y persistente del servidor, además ofrece las facilidades para la comunicación de los objetos del mundo virtual de Wonderland. Originalmente, Darkstar fue concebido como una plataforma para el desarrollo de juegos en línea, mundos virtuales y redes sociales.

jVoiceBridge [20] utiliza Voz sobre Protocolo de Internet (Voz sobre IP), provee a Wonderland de alta fidelidad en sonido, mezcla individual de canales de audio y audio estéreo en tiempo real. Este proyecto está basado en los estándares SIP & RTP por lo que puede ser integrado a la red telefónica convencional. jVoiceBridge facilita la inmersión auditiva en el ambiente virtual brindando una gama de calidades de voz tales como la presencial (atenuada a medida que se aumenta la distancia con el ente emisor), la telefónica, o la calidad de CD. Originalmente, jVoiceBridge fue desarrollado para permitir tareas como audio-conferencias, *chats* de voz y audio para ambientes virtuales 3D.

jMonkeyEngine (JME) [23] es un motor de renderizado que usa OpenGL (*Open Graphics Library)* y OpenAL (*Open Audio Library*) a través de LWJGL (*Lightweight Java Gaming Library*). JME es una arquitectura basada en grafos de escenas organizadas como árboles, esta organización permite descartar rápidamente ramas del árbol para el renderizado rápido de imágenes complejas. JME soporta un gran número de objetos activos; permite importar tanto imágenes como modelos de diferentes formatos; integrar aplicaciones desarrolladas como Java Applets, en AWT y en Swing. Su funcionalidad permite la implementación de juegos con calidad profesional.

El Proyecto Looking Glass [22] permite la interacción mejorada de aplicaciones de escritorio gracias al uso de ventanas y visualización 3D. En la actualidad, solo integra aplicaciones lanzadas desde plataformas Linux.

El proyecto Wonderland está en fase de desarrollo alfa (versión 0.5). Su versión 0.4 soporta aplicaciones compartidas 2D convencionales donde todos los usuarios tienen la misma visión de la aplicación y establecen un protocolo de trabajo por turnos. En la versión 0.4 es posible extender el mundo virtual mediante la creación de escenas, objetos estáticos y aplicaciones colaborativas tales como la pizarra interactiva o el visualizador PDF. La versión 0.5 pretende aumentar las capacidades de escalabilidad actuales (25 usuarios) mediante el uso de servidores de federación, permitir asignar diferentes comportamientos a los objetos mediante *scripting* y mejorar sustancialmente la expresividad de los avatares. A través de la página principal del proyecto [16] se accede al código de la versión 0.4 para usuarios y a la 0.5 para desarrolladores, se accede a tutoriales bien organizados para las diferentes versiones, y a vídeos que demuestran las diferentes características del proyecto. Existen además foros, wikis, blogs con mucha actividad de la comunidad que apoya este proyecto.

V. ENTORNOS EDUCATIVOS DESARROLLADOS SOBRE LAS PLATAFORMAS ANALIZADAS

Según D. Livingstone y J. Kemp [24], las actividades de enseñanza en plataformas como las analizadas, se pueden agrupar en las siguientes categorías:

- Simulación y escenificación de situaciones.
- Trabajo en grupo.
- Eventos y presentaciones.
- Actividades de construcción de objetos.

Utilizamos el marco establecido por D. Livingstone y J. Kemp para catalogar las experiencias educativas más significativas desarrolladas en las plataformas presentadas.

A. Simulación y escenificación de situaciones.

Las aplicaciones que simulan y escenifican situaciones requieren de una representación lo más fiel posible del objeto y la situación de estudio. La actividad consiste en diseminar información, observar el objeto o situación planteada y la manipulación del objeto para que el estudiante tenga la sensación de "estar ahí". En una simulación, las reglas y restricciones de la vida ordinaria están temporalmente suspendidas y reemplazadas por el conjunto de reglas operativas dentro del espacio y el tiempo donde se desarrolla la simulación.

En Second Life se han desarrollado diversas simulaciones para el aprendizaje de ciencias [25], por ejemplo, Genome Island es un entorno de aprendizaje experimental donde el estudiante puede interactuar con réplicas de experimentos genéticos clásicos, repetirlos, obtener datos, recibir instrucciones y también realizar evaluaciones. Otro ejemplo de simulación es The Herat Murmur Sim [26], la aplicación proporciona un espacio de entrenamiento donde los participantes escuchan el ritmo de los corazones de seis pacientes y hacen un diagnóstico para cada caso.

La empresa consultora de tecnología VEGA [27] utilizó Wonderland para desarrollar una academia virtual donde el aprendizaje puede llevarse a cabo de forma colaborativa. VEGA integró a la academia virtual, un simulador de avión desarrollado como una aplicación independiente.

El proyecto Arts Metaverse [28] desarrollado por la Universidad de British Columbia, simula ambientes virtuales 3D basados en Croquet. La aplicación permite a los estudiantes visitar edificios, comunidades, y culturas sin salir de su hogar o escuela (ver Fig. 1). De la exploración del ambiente virtual y la interacción con sus compañeros, el estudiante construye su propia comprensión de la arquitectura, la cultura, o la sociedad objeto de estudio.

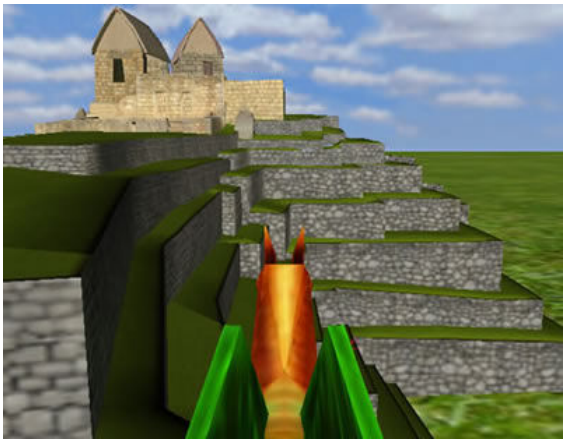


Fig. 1. Reconstrucción de Machu Picchu en Croquet. Universidad British Columbia. Proyecto: Arts Metaverse.

B. Trabajo en grupo.

Aquí se incluyen las aplicaciones donde es imprescindible la sincronización de varios colaboradores para el acceso simultáneo a un objeto virtual. Los participantes llevan a cabo tareas altamente colaborativas que incluyen planificación de actividades, tormenta de ideas, competición, negociación. Las tareas deben ser persistentes, por ello en estas aplicaciones se esperan actividades tales como el establecimiento de agenda de trabajo, la asignación de responsabilidades, y la posibilidad de retomar trabajo previo. Los protocolos sociales de comunicación son formales y pueden incluir juegos de rol. Este tipo de aplicaciones debe fomentar la alta cohesión entre participantes promocionando incluso actividades fuera del ambiente virtual. La comunicación debe ser lo más real posible.

Sun Microsystems promovió el proyecto *Wonderland for Kids* [29] entre dos grupos de niños de segundo grado de primaria en Freemont (California) y en Santiago de Chile. La experiencia pretendía la mejora de destrezas lingüísticas en un segundo idioma mediante juegos como "tres en raya".

Sobre Second Life encontramos aplicaciones que combinan las ventajas de inmersión y colaboración en la adquisición de destrezas y conocimiento. En este ámbito podemos citar la aplicación desarrollada por Thomson Netg [30] que ofrece entrenamiento para adquirir destrezas en el campo de los negocios y en el servicio de ventas.

El entorno 3D de Second Life también está siendo integrado con el LMS Moodle en el proyecto Sloodle [24], única aplicación conocida para la gerencia del aprendizaje. Sloodle recrea entornos de *elearning* tradicionales con espacios de aprendizaje inmersivos virtuales (ver Fig. 2).

Fig. 2. Proyecto Sloodle desarrollado en Second Life.

MPK20 [31] y Qwaq Forums [32] son proyectos de naturaleza netamente comercial que incluimos aquí dado que su funcionalidad tiene aplicación directa en entornos de aprendizaje.

El proyecto Espacio Virtual de Trabajo (MPK20) está siendo desarrollado por Sun Microsystems sobre Wonderland, permite la creación de espacios virtuales 3D donde pueden llevarse a cabo reuniones de trabajo, los usuarios pueden compartir documentos y reunirse con colegas. El proyecto enfatiza la utilización de audio de alta fidelidad, brinda además la posibilidad de realizar trabajo colaborativo sobre aplicaciones de escritorio 2D estándares.

Una experiencia semejante a la anterior es Qwaq Forums [32] desarrollada por Qwaq Inc. en Croquet. En esta aplicación es posible crear mundos virtuales para negocios (oficinas, salas de reuniones) y se permite a los usuarios trabajar y colaborar entre sí para identificar y resolver problemas.

C. *Eventos y presentaciones.*

Este tipo de aplicaciones se caracteriza por la construcción de un escenario similar al real donde los usuarios interactúan entre si de manera síncrona, con una colaboración de uno-a-muchos. El grupo se forma con individuos que están en lugares distantes y la dimensión del grupo puede llegar a ser grande.

En Second Life han sido llevadas a cabo diversas experiencias de este tipo, entre ellas vale la pena mencionar el Campus del New Media Consortium (NMC) [33]. Comprende eventos, clases, demostraciones e incluso exhibiciones de aprendizaje. Conferenciantes de la talla de Howard Rheingold, Henry Jenkins, y Daniel Reed han impartido charlas en el Campus del NMC. Este Campus ha sido además, sede de sesiones de conferencias tales como NYLC Nacional Service-Learning Conference, TCC Online Conference y EDUCAUSE Focus Session on Immersive Learning Environments.

La experiencia educativa más interesante realizada sobre Wonderland es sin duda, el proyecto MiRTLE (*A Mixed Reality Teaching and Learning Environment*) [34]. MiRTLE provee un entorno donde se integran el mundo real y el virtual para brindar la posibilidad a estudiantes de localidades remotas, participar en clases junto con estudiantes que asisten físicamente a las clases. Mientras el profesor imparte la clase puede interactuar tanto con los estudiantes presentes como con los avatares que representan a los estudiantes de localidades remotas. MiRTLE está en proceso de evaluación, la experiencia piloto se realiza en el Network Education College de Shanghai Jiao Tong University (SJTU). Se imparten clases en lugares como Shanghai, regiones de China tales como el Tibet, Yan'an, Xinjiang, y Ningxia. Los clientes pueden ser PCs, laptops, PDAs, IPTV, y teléfonos móviles. El objetivo a largo plazo del proyecto MiRTLE es crear todo un Campus con la integración de los mundos real y virtual (Ver Fig. 3)

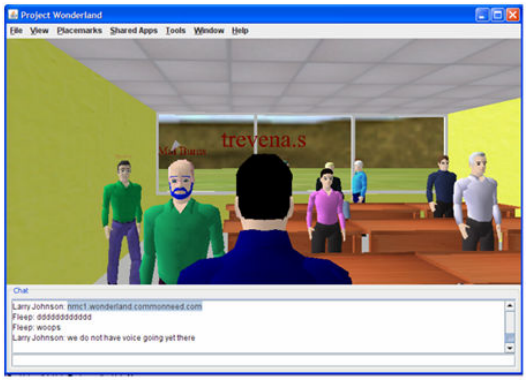


Fig. 3. Proyecto MiRTLE desarrollado en Wonderland.

D. *Actividades que involucran la construcción de objetos.*

En este tipo de aplicaciones, se espera que las tareas sean planificadas, que haya información acerca de cómo construir los objetos del mundo virtual, que haya un trabajo previo de preparación de escenario y de los objetos básicos de trabajo. El protocolo social puede ser tanto formal como informal y es necesario que los participantes estén suficientemente entrenados en el uso de la tecnología.

El proyecto educativo Greenbush Edusim [35] desarrollado en Croquet, pertenece a esta categoría de aplicaciones. Edusim es un entorno virtual 3D diseñado para ser utilizado en aulas de clase provistas de pizarras digitales interactivas conectadas entre sí. A través de las pizarras, los participantes integran, en un entorno preconstruido, los objetos que crean utilizando un menú de herramientas.

Tanto Wonderland como Second Life tienen las facilidades para la construcción colaborativa de objetos, sin embargo no se han encontrado aplicaciones educativas que exploten esas posibilidades.

Las aplicaciones reseñadas en este apartado, intentan facilitar el aprendizaje explotando el medio 3D. Sin embargo, se trata de experiencias con limitaciones en cuanto a la representación fiel del objeto de estudio, en cuanto a sus capacidades de colaboración y a sus posibilidades de gestionar entornos educativos. Los objetos educativos apropiados que se incluyen, son adaptaciones obvias de los hallados en entornos 2D.

VI. Conclusiones y Trabajo Futuro

El detenido estudio de las posibilidades que a día de hoy ofrecen tres de las más importantes plataformas 3D del mercado permite afirmar que dichas plataformas parecen estar listas para empezar a explotarse educativamente de formas que vayan más allá de lo meramente exploratorio y experimental. O dicho con otras palabras: aunque aún con amplio margen para la mejora, la capa tecnológica parece ser ya lo suficientemente sólida como para empezar a trabajar con garantías en la exploración y construcción de otros

niveles que, si bien sustentados sobre dicha capa tecnológica, se acerquen más al empleo docente de dicha tecnología y a la formulación de estrategias didácticas que adapten a estos entornos los mecanismos de colaboración y aprendizaje de los sistemas no-3D, y que a partir de ellos definan otros nuevos especialmente concebidos para aprovechar al máximo las nuevas y potentes características del 3D.

De esa manera, además, se cerrará la espiral del desarrollo tecnológico, al permitir que de alguna forma los nuevos dispositivos técnicos estén influenciados e incluso hasta cierto punto modelados por el empleo que los usuarios (docentes y discentes en este caso) den a la tecnología 3D en el campo educativo.

En esa línea de investigación se enmarca el proyecto del que este trabajo inicial forma parte. El análisis realizado en los apartados anteriores pone las bases para poder definir nuevos escenarios didácticos, nuevas formas de colaboración y nuevas vías de aprendizaje que se apoyen sobre un sustrato tecnológico cada vez más sólido pero que al mismo tiempo obliguen a la adaptación y a la reconfiguración parcial de dicho sustrato a fin de obtener mediante su empleo mejoras palpables en el aprendizaje.

Los servicios que a día de hoy ofrecen las plataformas analizadas, y sus perspectivas de evolución a corto plazo, ponen ante nosotros un potencial educativo creciente. Al menos eso puede deducirse si se comparan los servicios que ofrecen con los requisitos expresados al principio de este artículo. Ante la emergencia de este nuevo escenario educativo, se habrá de adoptar una actitud sistémica, holística, que partiendo de los entresijos tecnológicos, y sin perderlos nunca de vista, nos permita construir sobre los cimientos de dicha tecnología un robusto edificio de principios pedagógicos sólidos y acciones didácticas efectivas. Un edificio que sólo podrá alcanzar las dimensiones que sus cimientos prometen si se aborda su diseño y construcción desde una perspectiva pluridisciplinar en la que la creatividad bien concebida ha de jugar un papel fundamental.

No en vano nos toca asumir el papel de arquitectos de un nuevo mundo, y pocas veces será mejor empleada dicha expresión, en el que la voz 3D, los avatares con expresión corporal, los escenarios geográficos detallados, la visualización 3D de documentos y el resto de servicios ofrecidos por las plataformas actuales cobren vida más allá del movimiento, convirtiéndose en las piezas de un complejo puzzle con entidad propia. Un todo didáctico más allá de sus partes y herramientas concretas.

¿Practicar inglés en una Trafalgar Square virtual? ¿Espacios hipertextuales 3D abiertos a la exploración? ¿Aulas en las que organizar representaciones colaborativas en vivo? Éstas son algunas de las puertas que, con este trabajo, nos preparamos para abrir.

A. Trabajo Futuro.

Dentro del laboratorio de *elearning* Gradient [36] para entornos virtuales, trabajamos con la plataforma Wonderland. El primer requisito establecido para la elección de la plataforma es disponer sin restricciones del código fuente. SecondLife por tanto queda descartado.

Desde el punto de vista de los requisitos de las plataformas, de una forma u otra todas cumplen con los requisitos de colaboración descritos en [2]. Por eso para la evaluación se tomaron principalmente criterios técnicos y estratégicos: estabilidad, perspectivas de futuro, facilidad de desarrollo.

Croquet destaca por sus capacidades de creación dinámica de elementos en el mundo y por su capacidad para conectar mundos diferentes mediante portales. Por contra el mundo de Wonderland 0.4 es mucho más estático: Existe una gran separación entre el desarrollo de elementos y su uso. Sin embargo su estabilidad es mucho mayor. El ritmo de desarrollo es muy estable y el hecho de estar soportado por una compañía como SUN proporciona cierta seguridad respecto a su futuro. Su versión 0.5 además soluciona los problemas de creación dinámica de objetos. Otras características como la magnífica integración del audio, a través de SIP, o la capacidad de integrar aplicaciones de escritorio y el uso de lenguaje Java influyeron también en la elección.

Asi pues, tomando como base la plataforma Wonderland, los frentes de trabajo abiertos son los siguientes:

- Crear los objetos educativos básicos necesarios que servirán como base para construir servicios más complejos y elaborados: NPC's (non-player character), ambientes vivos, evaluadores...
- Facilitar al docente las herramientas para poner a disposición del alumno los componentes educativos necesarios en cada momento.
- Incorporar elementos de diseño instruccional estandarizado.
- Interacción mundo virtual/mundo real.

Como ejemplo concreto actualmente trabajamos en el desarrollo de un entorno de aprendizaje de la lengua española para extranjeros dentro del marco del proyecto España Virtual. Este entorno se basa en la recreación de lugares del mundo real (como calles, plazas o monumentos de ciudades españolas históricas) dentro del mundo virtual. Así se conectan los elementos lingüísticos con elementos culturales y se aumenta el grado de inmersión.

Una de las ideas principales sobre las que se está trabajando tiene que ver con el apartado de evaluación. Junto con los métodos convencionales utilizados por los docentes para evaluar la progresión de los alumnos, como puede ser la realización de ejercicios y preguntas preestablecidas, estamos explorando nuevas formas de evaluación basadas en la observación de las interacciones que llevan a cabo los alumnos con los objetos y personajes que se encuentran en el mundo virtual. Con este objetivo, se pretende desarrollar un sistema de registro que permita el almacenamiento, y posterior visualización y análisis, de las actividades realizadas por los alumnos dentro de este escenario 3D.

AGRADECIMIENTOS

Este trabajo ha sido realizado en el marco del proyecto “España Virtual”. España Virtual es un proyecto de I+D, subvencionado por el CDTI dentro del programa Ingenio 2010, orientado a la definición de la arquitectura, protocolos y estándares del futuro Internet 3D, con un foco especial en lo relativo a visualización 3D, inmersión en mundos virtuales, interacción entre usuarios y a la introducción de aspectos semánticos, sin dejar de lado el estudio y maduración de las tecnologías para el procesamiento masivo y almacenamiento de datos geográficos.

Con una duración de cuatro años, el proyecto está liderado por DEIMOS Space y cuenta con la participación del Centro Nacional de Información Geográfica (IGN/CNIG), Grid Systems, Indra Espacio, GeoVirtual, Androme Ibérica, GeoSpatiumLab, DNX y una decena de prestigiosos centros de investigación y universidades nacionales.

REFERENCIAS

[1] R. Bardolet. "La segunda estrella a la derecha. Informe Especial: Educación en Mundos Virtuales 3D", *Learning Review Latinoamérica*, Available: http://www.learningreview.com/informes-especiales-lr/educacion-en-mundos-virtuales-3D/15.html, [Accessed: November, 2008].

[2] Ch. Bouras and E. Giannaka and Th. Tsiatsos, "Virtual Collaboration Spaces: The EVE Community". *Symposium on Applications and the Internet*, pp. 48—55, 2003.

[SkypeA] Skype, "Skype", *Skype*. Available: http://www.skype.com/intl/es/, [Accessed: April, 2009].

[3] You Tube, "You Tube ", *You Tube* , Available: http://www.youtube.com, [Accessed: April, 2009].

[4] Flickr, "Flickr", *Flickr*, Available: http://www.flickr.com/, [Accessed: April, 2009].

[FACEBOOKA] Facebook, "Facebook", Facebook. Available: http://www.facebook.com/, [Accessed: April, 2009].

[5] Immersive Education, "Immersive Education Initiative", *Immersive Education*, Available: http://immersiveeducation.org/ , [Accessed: April, 2009].

[6] Linden Lab, "SecondLife", *Linden Lab*, Available: http://secondlife.com/. [Accessed: April, 2009].

[7] Linden Lab, "Linden Lab", *Linden Lab*, Available: http://lindenlab.com/, [Accessed: April, 2009].

[8] D.A. Smith and A. Kay and A. Raab and D.P. Reed, "Croquet – A Collaboration System Architecture", *First Conference on Creating, Connecting and Collaborating through Computing*, c5, pp.2, 2003.

[9] The Croquet Consortium, "The Croquet Consortium", *The Croquet Consortium*. Available: http://www.opencroquet.org/index.php/Main_Page. [Accessed: April, 2009].

[10] A. Black and S. Ducasse and O. Nierstrasz and D. Pollet and D. Cassou and M. Denker, "Squeak by Example", Square Bracket Associates, 2007.

[11] Squeak, "Squeak", *Squeak*, Available: http://www.squeak.org/. [Accessed: April, 2009].

[12] J. Maloney, "An Introduction to Morphic: The Squeak User Interface Framework", In: *Squeak: Open Personal Computing and Multimedia*, Prentice Hall, 2002, pp. 39—67.

[13] eToys, "Squeakland. Home of Squeak Etoys", *eToys*, Available: http://www.squeakland.org/. [Accessed: April, 2009].

[14] D.P. Reed, "Implementing Atomic Actions on Decentralized Data", *ACM Transactions on Computer Systems*, Vol. 1, pp. 3—23, 1983.

[15] The Croquet Consortium, "Open Cobalt. Virtual workspace browser and toolkit", *Cobalt*, Available: http://www.duke.edu/~julian/Cobalt/Home.html. [Accessed: April, 2009].

[16] Sun Microsystems, "Project Wonderland: Toolkit for Building 3D Virtual Worlds", *Sun Microsystems Labs. Wonderland Project*. Available: https://lg3d-wonderland.dev.java.net/. [Accessed: April, 2009].

[17] Sun Microsystems, "Sun Microsystems Laboratories", *Sun Microsystems Labs.* Available: http://research.sun.com/. [Accessed: April, 2009].

[18] Sun Microsystems, "The Source for Java Developers", *Sun Microsystems.* Available: http://java.sun.com/. [Accessed: April, 2009].

[19] Sun Microsystems, "Project Darkstar. Open Source for the Online Game Universe". *Sun Microsystems Labs.* Available: http://www.projectdarkstar.com/ . [Accessed: April, 2009].

[20] Sun Microsystems, "jVoiceBridge". *Sun Microsystems Labs.* Available: https://jvoicebridge.dev.java.net/. [Accessed: April, 2009].

[21] Sun Microsystems, "jMonjeyEngine. Serious monkeys. Serious engine". *Sun Microsystems Labs.* Available: http://www.jmonkeyengine.com/. [Accessed: April, 2009].

[22] Sun Microsystems, "Welcome to Project Looking Glass!". *Sun Microsystems Labs.* Available: https://lg3d.dev.java.net/. [Accessed: April, 2009].

[23] Sun Microsystems, "jMonjeyEngine. Features List". *Sun Microsystems Labs.* Available: http://www.jmonkeyengine.com/wiki/doku.php?id=complete_features_list. [Accessed: April, 2009].

[24] D. Livingston and J. Kemp, "Integrando entornos de aprendizaje basados en Web y 3D: Second Life y Moodle se encuentran", *Novática*. N. 193,pp. 7—12, 2008.

[25] Linden Labs, "Science Learning Opportunities in Second Life", You Tube, Available: http://www.youtube.com/watch?v=EfsSGBraUhc&feature=related, [Accessed: April, 2009].

[26] J. Kemp, M. Adamant y E. Pasteur, "The Herat Murmur Sim", *Second Life*, Available: http://slurl.com/secondlife/waterhead/130/37. [Accessed: April, 2009].

[27] Vega, "Vega", Vega, Available: http://www.vega-group.com/aboutus/, [Accessed: April, 2009].

[28] University of British Columbia, "Arts Metaverse", *University of British Columbia*, Available: http://artsmetaverse.arts.ubc.ca/, [Accessed: April, 2009].

[29] Sun Microsystems, "Wonderland for Kids", *Sun Microsystems*, Available:http://blogs.sun.com/wonderland/entry/wonderland_with_kids. [Accessed: April. 8, 2009].

[30] Business Communicators of Second Life. "Thomson NetG Second Life Corporate Training Campus", Available: http://freshtakes.typepad.com/sl_communicators/2006/09/thomson_netg_se.html. [Accessed: April, 2009].

[31] Sun Microsystems, "MPK20: Sun's Virtual Workplace", *Sun Microsystems*, Available: http://research.sun.com/projects/mc/mpk20.html. [Accessed: April, 2009].

[32] Qwaq, "Qwaq Forums", *Qwaq*, Available: http://www.qwaq.com/, [Accessed: April, 2009].

[33] New Media Consortium Campus, "New Media Consortium Campus", Linden Labs, Available: http://slurl.com/secondlife/NMC%20Campus/138/225/43, [Accessed: April, 2009].

[34] V. Callaghan and M. Gardner and B. Horan and J. Scott and L. Shen and M. Wang, "A Mixed Reality Teaching and Learning Environment". ICHL 2008, pp. 54--65.

[35] Edusim, "Edusim – 3D virtual words for the classroom interactive whiteboard", *Cobalt*, Available: http://edusim3d.com/, [Accessed: April, 2009].

[36] Gradient, "Gradient", Gradient, Available: http://gradient.it.uc3m.es/, [Accessed: April, 2009].

LA LEY DE MOORE Y EL VÉRTIGO SOCIAL

Pedro Costa Morata(*), Beatriz Moreno Llorente(**), Eloy Portillo Aldana(*)
(*) Departamento de Ingenierías y Arquitecturas Telemáticas (DIATEL)
Escuela Universitaria de Ingeniería Técnica de Telecomunicación (EUITT)
Universidad Politécnica de Madrid (UPM)
Ctra. de Valencia, Km. 7. 28031 Madrid
(**) Canal Satélite Digital S. L.
pcosta@diatel.upm.es, bmoreno@sogecable.com, portillo@diatel.upm.es

Resumen- Este artículo pretende llamar la atención sobre los inocultables efectos que la "carrera por la velocidad" produce en lo personal y lo social. Se alude, concretamente, a la velocidad electrónica, tomando como punto de partida la conocida como "Ley de Moore", que expresa en términos exponenciales una triple relación directa entre los avances en velocidad de computación, la progresiva miniaturización de componentes y sistemas y su constante abaratamiento. Los autores salen al paso de las dificultades que se alzan frente a estas pretensiones, más allá de su aparente cumplimiento durante decenios, y centran su análisis en las repercusiones ambientales y en los efectos negativos de la "aceleración social" que impone esa conquista continua de la velocidad tecnológica, concretamente electrónica.

***Palabras Clave*- Ley de Moore, Revolución de las TIC, Efectos socio-económicos de la automatización, circuitos integrados, ciencias sociales**

I. DEFINICIONES Y SIGNIFICADOS

La llamada Ley de Moore, de aplicación al campo de la electrónica (LM), se expresa así: "Cada 18 meses, aproximadamente, se duplica el número de transistores en un circuito integrado". Se data su origen en 1965 y se cita como referencia un artículo publicado por Gordon E. Moore en la revista Electronics [1]. Hacia 1975, este mismo autor corrigió el primer enunciado para "rebajarlo" a una duplicación de capacidad operacional cada dos años, y en 2007 anunció que "su ley" dejaría de cumplirse en 10 ó 15 años.

De todas formas, es importante subrayar que lo esencial de los enunciados de Moore (importante directivo de los laboratorios de Fairchild Semiconductors en el momento de la "definición" de esa ley) en el artículo citado se refiere a los costes de los circuitos integrados de semiconductores y a su proyección hacia el futuro, observando en concreto que "la complejidad para los costes mínimos por componente se ha incrementado en una relación de dos cada año". Las formulaciones usuales resultan, así, deducciones de los contenidos y observaciones de Moore en ese artículo, en el que, insistimos, su preocupación se centra en los costes de fabricación y su evolución futura.

Poco importa, en todo caso, la fidelidad literal con que se expresa una tan famosa ley en la electrónica de los últimos 40 años, ya que, en lo esencial, lo que nos transmite, con veracidad incuestionable, es el incremento sostenido de la velocidad de operación (número de componentes semiconductores en un circuito), la reducción del tamaño y el abaratamiento de los costes de producción. Por otra parte, el mismo autor se refiere a ella como "observación" y, como mucho, "ley empírica".

La LM ha ido, en lo sustancial, cumpliéndose desde los años de 1950, es decir, a partir de que se generalizara el uso de los semiconductores tras la puesta a punto del transistor (1947)[1], del circuito integrado (1961) y del microprocesador (1971). Concretamente, Moore partió de 1959 como "año cero" para su análisis, tomando como referencia un transistor por circuito simple; y observó que hacia 1965 eran 64 transistores los que habían podido acumularse en un mismo circuito (deduciéndose, en consecuencia, que la complejidad de los circuitos electrónicos se duplicaba cada año…).

Los desarrollos científico-tecnológicos posteriores, digamos, a los años de 1940/50– se han orientado hacia la eficacia funcional de los procesos tecno-económicos, viniendo la velocidad de operación de la mano de la nueva electrónica de estado sólido, es decir, de los semiconductores. La LM es una realidad –si bien de funcionamiento no exacto ni de orden "natural"– puramente instrumental y de intermediación: por una parte refleja impulsos y dinámicas subyacentes no estrictamente científico-técnicas y por otra supone un paso o etapa –importante, si bien no esencial– en el intenso despliegue de la electrónica y las telecomunicaciones, sobre todo las orientadas a la guerra, la eficiencia productiva y el consumo de masas.

La LM refleja un hecho típica e indisolublemente científico-técnico, ya que las técnicas que permiten incrementar la velocidad de operación al tiempo que la reducción del tamaño de los componentes, sistemas y

[1] El hito fundamental en la historia tecnológica de los semiconductores es, efectivamente, la realización del primer transistor por Bardeen, Brattain y Shockley en los Laboratorios Bell de la AT&T; los tres fueron galardonados con el Nóbel de Física en 1956.

equipos se basan, o se acompañan, de un intenso proceso investigador sobre las propiedades de la materia. Y, concretamente, pretende ligar tres factores, o variables no homogéneas –velocidad, espacio, coste– que reducen su valor de forma directa y simultánea, lo que puede parecer excesivo y debe mover a afinar el análisis y a tener en cuenta elementos ocultos o minusvalorados.

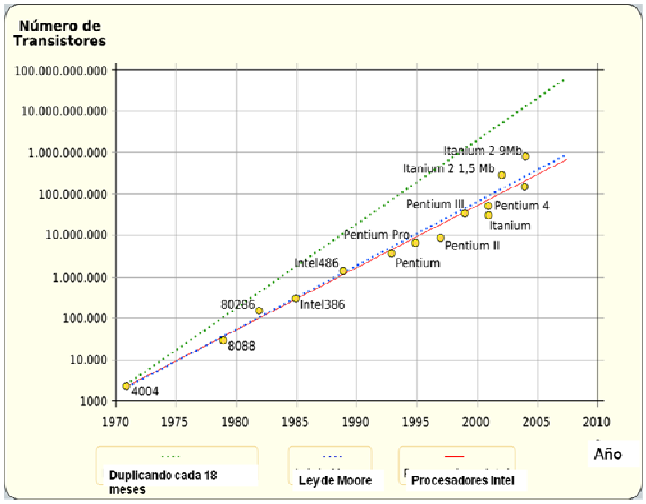


Fig. 1. Ley de Moore

En todo caso, tres son las consideraciones iniciales que consideramos oportunas plantear:

a) Como ley, la LM traslada sus contenidos desde la velocidad a la miniaturización, lo que es aproximadamente equivalente; y de ahí a la reducción de costes en la forma de un corolario vinculado de forma distinta a las dos primeras variables. Es, sin embargo, la cuestión de los costes lo que debe llamarnos la atención, o su equivalente: el que los precios bajen al mismo tiempo que suben las prestaciones.

b) Como indicador social (y sociológico) la LM sugiere realidades más allá de lo aritmético y, además, resulta de capacidad múltiple ya que pretende, en una primera ampliación, expresar la correlación técnica-economía, siendo ambas variables de índole bien distinta (una es material en gran medida, la otra es social).

c) Singularmente, la LM forma parte de la llamada "revolución digital", o contribuye eficazmente a constituirla. Pero acerca de este concepto de revolución digital mucho hay que decir, y no todos los análisis coinciden en el análisis de los elementos y supuestos de esta nueva revolución (a la luz de la cadencia de revoluciones tecnológicas que en el mundo y la historia ha tenido lugar).

En este trabajo no podemos ignorar que la LM –su "cumplimiento"– subyace a tendencias históricas negativas, convertidas progresivamente en hechos dramáticos habidos en los 30 últimos años en la estructura socioeconómica del mundo "desarrollado"; y que su relación con estas tendencias y estos hechos debe darse por consistente.

Y en relación con el principal problema que aquí consideraremos, la velocidad tecnológico-electrónica, se ha de inscribir en el persistente análisis del arquitecto, pensador y crítico tecnológico Paul Virilio, considerado "el filósofo de la velocidad", que llega a calificar esta aceleración tecnológica como una contaminación dromosférica (de dromos: carrera veloz) y reivindica la importancia de una Ecología gris que atienda al amplio espectro de problemas –psicológicos, sociales y también ambientales– que se desprenden de "la contaminación de las distancias y de las magnitudes de tiempo"[1], en definitiva, de la reducción de la dimensiones de lo natural. El gris es el color que caracteriza a la tecnología y, sobre todo, a los instrumentos y sistemas dromológicos, que tienen que ver con la velocidad, pero sobre todo hay que tener en cuenta que "la velocidad mata al color: el giróscopo, cuando gira con rapidez, forma el gris"[2].

Aunque no se puede ignorar que la crítica de la velocidad y de la dinámica de la aceleración que conlleva la civilización tecnológica está presente en numerosos intelectuales y pensadores anteriores, desde luego, a Virilio, y aquí debemos citar a Ivan Illich, quien con su célebre Energía y equidad ya advirtió del absurdo que encierra en sí misma la persecución de la velocidad mecánica: "Acrecentando el medio aleja el fin"[3].

II. El entorno global de la "vigencia" de la LM

El entorno a considerar en el análisis de la LM ha de ser, al menos, doble: histórico y económico; y se extiende en un primer momento desde la aceleración hacia la miniaturización, ampliándose como corolario al abaratamiento. Son pues tres las dimensiones técnicas del proceso en un sistema de –al menos– tres ejes coordenados: el tiempo, el espacio y el coste.

Por lo que al "fondo histórico" se refiere la "aceleración" vivida por la electrónica, trasladada al ámbito de las telecomunicaciones, se ha de inscribir en la evolución histórica de la velocidad material del transporte y la comunicación, que por lo que a la comunicación se refiere presenta dos puntos singulares de "aceleración": la aparición del telégrafo (óptico primero, eléctrico poco después en el intervalo de una treintena de años, entre 1794-1825) y la aparición de la radio (primeros ensayos de 1888, con Hertz, y primeras aplicaciones comerciales-militares hacia 1910, con Marconi). Al alcanzarse en la práctica la velocidad de la luz en las radiocomunicaciones, se ha llegado a unos límites aparentemente estables que, desde el punto de vista económico e industrial, podría parecer que limitaban claramente esos dos campos de expansión.

Es importante observar que la LM se cumple –o se observa, como modestamente diría el propio Moore– atravesando dos etapas históricas que desde el punto de vista socio-económico-político-militar aparecen claramente separadas, aunque se hayan sucedido sin solución de continuidad: la keynesiana de la Guerra fría y la neoliberal del hegemonismo norteamericano. Y por lo que a esta

[1] Citado por S. Rial, *Paul Virilio y los límites de la velocidad*, [2] p. 33.
[2] En expresión de Paul Morand (citado por Rial, *op. cit.*, p. 32.
[3] Ivan Illich, *Energía y equidad*, [3] p. 59.

segunda fase se refiere (que se inicia en la década crítica de 1970), subyacen de hecho tendencias nefastas, que han llevado a la progresiva sujeción, explotación y humillación del ciudadano-trabajador (no escapándose el técnico, por cualificado que pretenda ser)[1].

Pero esta "indiferencia" ante el distinto discurrir del tiempo político-económico no debe hacernos pensar que esta LM sea independiente de ellos, o se exprese con autonomía, espontaneidad o vida propia, como propugnarían los partidarios del determinismo tecnológico. Por el contrario, la LM se expresa, siempre, como producto y reflejo de su época, que en materia electrónica presenta escasas variaciones ya que ciencia y tecnología son, permanentemente, la punta de lanza de un sistema económico que busca obsesivamente la rentabilidad y la competitividad por intermedio, fundamentalmente, de la reducción de los costes laborales (con los que se relaciona inversamente el proceso de aportación tecnológica a la producción).

Y en ambas etapas las consecuencias meramente técnicas de la LM se han distinguido, sobre todo, por su incidencia económico-empresarial, mejorando la organización interna de las empresas (racionalizándolas) y, por otro lado, incrementando la actividad y los beneficios. Esta es la característica básica histórica, económica y funcional, de la ciudad informacional, en los términos empleados por Manuel Castells, cuyo modelo es originalmente norteamericano pero que se ha extendido –llevado, precisamente, por su propia esencia tecnológica– a los países tecnológicamente desarrollados y a toda la estructura productiva internacional [5].

También atendiendo a su "ambiente histórico" la LM es un producto de los tiempos optimistas y desarrollistas, esencialmente cuantitativos: procede directamente de la etapa histórica en la que el más era en todo equivalente al mejor (y por lo tanto, más deseable), pero ésta es una fase histórica que inició su quiebra, irreversible, con la percepción global de la crisis ambiental.

Con respecto a las tendencias de tipo económico que subyacen en la LM la primera observación debe dirigirse hacia el tercer elemento, o variable, de naturaleza radicalmente distinta, quizás incoherente con los otros dos: si bien el tiempo y el espacio parece que pueden ser contemplados con propiedad (sin falsa apariencia, sin trampa), el coste, sin embargo, sugiere complejidad y una evaluación mucho más afinada que la que hace la industria, incluso el sistema económico. Si hemos de "acercarlo" a realidades tan nítidas (aunque tan simbólicas, al tiempo), el coste como concepto, realidad y símbolo no puede ser identificado ni evaluado por la voluntad, el método o los intereses de la industria.

Cabría indagar, profundizando, sobre la tendencia de fondo que subyace en este proceso veloz y que, como está bien claro, impregna toda la vida social; pero si dejamos de lado el móvil militar, singularmente presente en la historia de las técnicas de la telecomunicación y de la electrónica en general, no cabe duda de que es la pulsión económica lo que mueve al desarrollo tecnológico (más aun que al científico).

Así, es evidente que esta velocidad de computación a la que alude la LM ha incrementado la eficiencia empresarial extraordinariamente (de dos formas, una interna y otra externa) y en este sentido no puede dudarse de su interés, utilidad y eficacia[2]. Podemos admitir, pues, que el entorno más amplio, casi diríamos global en el mundo de las llamadas "tecnologías de la información y la comunicación" (TIC), es de tipo económico-crematístico, y aquí reside lo sustancial de –no solamente los mecanismos de decisión– sino también de la planificación y la prospectiva[3].

Estas fuerzas de tipo económico que subyacen en el desarrollo de las TIC se han ido identificando más y más con las ambiciones del mercado y, de hecho, la red Internet viene a ser la imagen tecnológica y social del mercado abierto perfecto, teniendo como vocación y finalidad confesadas el intercambio universal y sin restricciones.

De todas formas, parece excesivo decir que "a medida que los componentes e ingredientes de las plataformas con base de silicio crecen en desempeño se vuelven exponencialmente más económicos de producir y, por lo tanto, más abundantes, poderosos y transparentemente integrados en nuestras vidas diarias"[4]. Porque está claro que un enunciado así ignora el complejo significado del coste económico, que muy pocas veces es simple (y menos, en la vida científico-tecnológica).

Especial interés posee la consideración de los costes que suscita la LM, ya que la reducción del coste que se proclama es lo que más posibilidades de análisis social y global nos sugiere. Así, en la evolución de los costes económicos se han de tener en cuenta:

1. Los costes monetarizables de forma convencional:
   a) Considerando los efectos directos
   b) Desvelando los costes colaterales
2. Los costes ocultos, o más difíciles de discernir y evaluar:
   a) Identificando los "nuevos" (o progresivos) costes propios
   b) Teniendo en cuenta los costes externos

Son, sin duda, los costes ocultos los que deben interesarnos si decidimos observar bien a fondo y sin detenernos en los aspectos que mejor conocemos o más nos interesan. Con toda probabilidad, Moore no ha captado el complejo extra-tecnológico de las numerosas repercusiones de su "ley" en el terreno de los costes ocultos, o no

[1] Por lo que se refiere a esta evolución de los últimos 30 años, e incidiendo en la pérdida de significación salarial de las clases trabajadoras y medias, ver: N. Burgi, Noëlle [4]

[2] El propio Moore se ha beneficiado de este efecto económico-crematístico evidentemente positivo, y la revista *Forbes* lo colocaba, cuando se jubiló a los 72 años (2001), en el número 60º de la lista de personas más ricas del mundo.

[3] La aversión tradicional de la doctrina liberal por la planificación muestra una vez más la relatividad con que se enuncia y proclama, ya que la industria de las TIC se basan esencialmente atendiendo a la previsión y la prospectiva.

[4] Ceccarell, Pablo: "¿Qué es la Ley de Moore?, en *Seguilaflecha.com* (4-4-2008).

comerciales, o al menos, su relación con recientes y cada vez mejor estudiadas circunstancias, de las que subrayamos éstas:

1. La velocidad social, en primer lugar. Es decir, el impulso global, y personal, a las prisas y el estrés, lo que acelera la realidad y tiende a hacerla desaparecer.
2. La pérdida del sentido analógico de las cosas (¡y de la realidad!).
3. La puja hacia el límite y el extremo, como los deportes de riesgo o muy competitivos.
4. La pérdida de la intimidad y la vigilancia global (ubicua, completa).
5. Las catástrofes y los miedos técnicos, que han adquirido una envergadura desconocida (los cambios informáticos del año 2000, las consecuencias de la alta vulnerabilidad/fragilidad de las redes complejas…).

III. Interpretación adicional sobre los límites y los efectos: los excesos de la velocidad electrónica

Desde hace tiempo tanto la ciencia como la tecnología (C-T) buscan sistemáticamente los límites y cultivan los excesos. No debe extrañar que se señale a la Guerra Fría como fase significativa en la configuración de una C-T de vértigo, de competencia y lucha, de búsqueda obsesiva, en definitiva, de resultados límite.

Esta C-T "extrema" gusta de ceñirse a eslóganes que aluden a evidentes límites y excesos: de forma paralela a como la enseña olímpica alude a "más rápido, más alto, más fuerte"1, la enseña de la C-T parece perseguir este otro, tan parecido, del "más rápido, más pequeño, más barato". Y éste parece ser definitivamente el lema de la mundialización, que es y pretende ser económico-financiera (y no otra cosa): "por una parte, la extrema reducción de las distancias, resultante de la compresión temporal de los transportes y las comunicaciones; por otra, la generalización en curso de la televigilancia"[2].

Pero en un mundo físico de limitaciones este eslogan resulta osado e incluso imprudente; y si se le llega a dar carta de naturaleza no ha de extrañar que, inevitablemente, lleve a la C-T a afrontar contradicciones de nada fácil solución.

Imitando al sistema capitalista –o como una imagen de éste, que puja siempre hacia los límites, sean del beneficio, del máximo aprovechamiento o del abuso– la C-T como fenómeno típico de la época histórica de recomposición de las relaciones económicas globales, tensa la cuerda y renuncia a imponerse límites. "Arrastrada durante casi medio siglo en la carrera armamentística de la era de la disuasión entre el Este y el Oeste, la ciencia ha evolucionado únicamente en busca de resultados límites, en detrimento del descubrimiento de una verdad coherente y útil a la humanidad"[3].

Y ya hemos señalado que el propio Moore admite correcciones y límites para "su" ley. Como "pegas" o puntos de quiebra de su enunciado o previsiones se anuncian éstos:

– El consumo eléctrico, la potencia exigida y los problemas de refrigeración que repercuten en variables importantes, como el peso y el volumen.
– El coste creciente de los equipamientos, a causa de su complejidad.
– Las limitaciones del aprendizaje y del conocimiento.
– La quiebra de los objetivos de fondo, o subyacentes, lo que implica cambios incluso de paradigma[4].
– Ante la necesidad ecológica de 'Reducir, Reutilizar, Reciclar', la carrera de la miniaturización la dificulta, ya que los componentes además de más pequeños tienen una vida menor y son más difíciles de reparar, reciclar.

Esos límites resultan directos y, por lo tanto, perceptibles y pertenecen al propio ámbito "regido" por la LM. Pero debemos insistir en otro tipo de límites, singularmente dos, que tienen que ver con el impacto ambiental (que, contra todo pronóstico, resultan cada día más significativos) y los relacionados con la velocidad precisamente, y sus repercusiones humanas y sociales.

Acerca del primero de esos límites (o claves para una mejor interpretación de éstos) hay que reconocer que es verdad que el mundo de la electrónica y de las telecomunicaciones (y la informática, como consecuencia) parece evolucionar eludiendo las limitaciones que la naturaleza impone a los procesos económico-productivos y las creaciones tecnológicas, pero también lo es que este mundo, o complejo tecno-científico, constituye una preocupación creciente en las sociedades desarrolladas actuales. En un mundo en el que se avistan límites, frenos y decepciones en todos los ámbitos de matiz económico-tecnológico el complejo mundo de la electrónica y las telecomunicaciones se ha beneficiado durante mucho tiempo de un aura de limpieza ambiental que, actualmente, ya ha perdido.

Este "prestigio" ha venido quebrándose al tiempo que se prestaba una mayor atención a diversos factores de creciente presencia en esta industria; tanto los metales tóxicos empleados como los importantes consumos energéticos y la creciente importancia de los procesos de tratamiento y reciclaje de los residuos electrónicos en general han ido destruyendo esa buena fama de la electrónica como industria limpia.

La segunda clave de interpretación de la presunta "vulneración" de límites por parte de la C-T apela al análisis de la velocidad en general y de la electromagnética en particular, que es la tarea en la que se ha especializado Virilio. Éste advierte sobre lo que esta "aceleración de la realidad" supone, relanzando la C-T de la desaparición, es

---

1 En realidad, *citius, altius, fortius*, es decir, "el más rápido, el más alto, el más fuerte" (1891).
2 P. Virilio: *La bomba informática*, [6] p. 23.
3 Virilio, *op. cit.*, p. 11.
4 Este cambio significaría que la velocidad de operación –o la búsqueda de la miniaturización– deja de ser objetivo básico en la investigación y la fabricación de componentes o equipos; y que se prima el ahorro energético, por ejemplo, o la estabilidad funcional de los sistemas, dándole durabilidad en lugar de sustituibilidad.

decir, de un saber más cibernético que enciclopédico, que niega toda realidad objetiva y que nos lleva a una estética de la desaparición científica.

Está claro que la velocidad parece surgir, de forma íntima e intrínseca, de la dinámica del desarrollo económico occidental, lo que a su vez muchos relacionan con un impulso humano, íntimo, de llegar antes y de comprimir crecientemente el tiempo y la distancia. Pero este rasgo, que se pretende típicamente humano, de la propensión a la rapidez no es fácil de demostrar si no se lo relaciona con la actividad económica y la competencia empresarial, y esto pertenece más a lo social-colectivo que a lo humano-personal.

De hecho, la historia demuestra que la "conquista" de la velocidad, tanto en los transportes como en las comunicaciones, ha permitido la concentración del poder. Virilio subraya que en esta carrera histórica de la velocidad siempre están presentes el poder y la guerra, bien por contribuir a resolver rivalidades, bien por emplearse a la conquista de territorios[1].

El marco de la velocidad nos lleva al análisis del vértigo y a la crítica de los excesos y de las tendencias al límite observables tanto en la ciencia como en la técnica. El vértigo y la prisa que ambas introducen en la vida ordinaria (y que la LEM certifica) nos sitúan directamente en ese marco de influencias obviamente existentes. Y de ellos se desprenden realidades y efectos que implican aumento y extensión del estrés social, como sucede con:

1. La presión por la sustitución de aparatos y programas, que supone una humillación para quienes no están habitualmente implicados en este mundo o en esta profesión. Esta "conminación" a cambiar bajo la presión de la –presunta– mejora significa la humillación en muchos casos…
2. El derroche por la permanente sustitución de equipos y software, que no solamente perjudican el trabajo pausado y productivo sino que contribuye a la ingente producción de desechos, con lo que esto impone en cuanto a reciclaje, descontaminación y gestión en general. Ningún derroche, o despilfarro, debe considerarse un acierto o una ventaja económica o social, y por lo tanto la tecnología que lo genera debe someterse a crítica radical.
3. El objetivo es producir y consumir, es decir, crematístico; y no es específicamente aportar bienes de interés social. Pero hay que reconocer que esta realidad es escasamente –o al menos, insuficientemente– percibida por la comunidad electrónica.
4. De forma más indirecta, pero no menos característica, esta velocidad y esta miniaturización inducen un individualismo aislacionista, una especie de "autismo tecnológico" que trastoca el ocio y su tiempo despojándolos de su atractivo social en un ejercicio personal de reducciones y que establece la "separación" respecto del entorno inmediato. Se trata, en definitiva, de deslindar todo lo que aleja, en el comportamiento y las relaciones personales y sociales, la "proximidad electromagnética"[2].
5. Además la tecnología no dura lo suficiente para que el técnico llegue a dominarla, sino que en estados todavía tempranos se le obliga a hacer un 'upgrade', un cambio de modelo con el consiguiente estrés para el técnico.

IV. La velocidad electrónica como valor discutible: propuestas para la reflexión

La velocidad electrónica no es un valor absoluto, sino relativo y particular, porque excede la mera significación electrónica. Es verdad que en lo técnico sí está bien circunscrito y posee innegable objetividad, pero su alcance global es mucho más amplio.

El eslogan aguafiestas de que "No hay almuerzo gratis"[3] debe aplicarse, también, a la industria electrónica, sin eludir el matizar sus logros en velocidad de operación e integración. Se trata, en esencia, de determinar –descubrir, desentrañar– dónde están los costes que hacen que ese "almuerzo" tan optimista a que invitan los espectaculares avances en rapidez y eficacia, no lo sea tanto, o nos recuerde que siempre habrá de ser pagado (por alguien, por algo, ahora o después).

La tendencia –acientífica, eufórica, excesiva– a extrapolar las ventajas directas de la velocidad electrónica a ámbitos distintos y más amplios, lleva a afirmaciones que, por repetitivas, se convierten en nuevas "leyes" de cumplimiento pretendidamente inexorable. Así, tanto el rendimiento informático creciente como la reducción progresiva de costes se convierten en enunciados reales pero inexactos y, sobre todo, tienen un inevitable fin. Y, desde luego, no son extrapolables al terreno de lo social, aunque esta "ampliación" se haya convertido en práctica habitual.

No es posible demostrar que de la LM se desprenda, por ejemplo, una mejora de la calidad de vida, o claros avances democráticos, o ni siquiera que se reduzca el tiempo de trabajo y se aumente el de ocio… Efectivamente, las estadísticas van confirmando que desde hace una par de décadas la jornada laboral tiende a aumentar precisamente en el sector servicios, y afecta muy sensiblemente al de la electrónica y las telecomunicaciones[4].

La historia, por otra parte, nos desencanta en relación con las pretensiones que el progreso ilustrado exhibía y que la Revolución industrial hizo suyas por la fuerza de un desarrollo económico tan desconocido como avasallador. No obstante, el culto al progreso se reaviva periódicamente[5],

[1] Virilio, *op. cit.,* p. 21.
[2] Rial, *op. cit*., p. 38.

[3] En su origen, este eslogan es de tipo económico y se atribuye al Premio Nobel de Economía, Milton Friedman; pero otros lo han relacionado con los problemas ambientales, indicando que cada proceso económico, incluyendo los muy eficientes o limpios, implican un cierto coste ecológico-ambiental.
[4] Los intentos europeo-comunitarios de elevar la jornada laboral máxima legal hasta las 65 horas no son una anécdota ni un "desvarío" de la Comisión europea, sino la concreción de esa tendencia, que es simultánea con incrementos extraordinarios de la productividad y de... la velocidad electrónica.
[5] Y no se quiere recordar que el primer golpe que se le infirió, aun antes de la exitosa formulación de Condorcet y del optimismo industrialista del siglo XIX, corrió a cargo del

cabalgando triunfante a lomos de las sucesivas "revoluciones" científico-tecnológicas que los tiempos vienen contemplando. No ha de extrañar que cuando la identificación entre velocidad y progreso supera todas las imprudencias de lo extrapolable, surjan movimientos sociales, aún testimoniales, que planteen todo lo contrario, es decir, que cualquier progreso humano y social ha estar vinculado con el discurrir pausado y el rechazo de las prisas: se trata del movimiento *slow*[1].

Es verdad que, en gran medida, el trabajo de identificación de los efectos sociales de la velocidad electrónica está pendiente de hacer, ya que sobre todo debiera realizarse desde dentro del mundo de la electrónica y las telecomunicaciones. La crítica "exterior" ha avanzado más, desde luego, pero sigue siendo principalmente global y no entra en las peculiaridades de los avances electrónicos ni en las fuerzas que los impulsan.

Nuestra propuesta básica al traer a consideración el tema de la LM –en este texto que a muchos puede parecer provocador– es principalmente de reflexión sobre algo –la velocidad electrónica– que aparece como muy positivo y productivo en el entorno de nuestra profesión y dedicación, pero que es necesario analizar vinculado a su entorno o marco de aplicación y desarrollo: digamos, en definitiva, que debemos ligarlo a "su circunstancia". A su circunstancia, precisamente, diría Ortega y Gasset [7], quien demostrando una vez más su enorme capacidad para afrontar los problemas que afectan a las sociedades humanas, por nuevos o "ajenos" a la filosofía que pudieran parecer, ya en 1933 no dudaba en señalar –en su Meditación de la técnica– que la técnica es "la producción de lo superfluo", aludiendo a la idea de necesidad y matizando la "suficiencia animal" de los humanos. (En realidad, el filósofo venía a lanzar una llamada a la necesaria recomposición psicológica y sociológica, filosófica en suma, a que los tiempos modernos nos obligan una y otra vez, y en concreto al afrontar la realidad del homo technologicus.)

Dice Virilio que el progreso técnico no resuelve los problemas de la humanidad, como mucho los desplaza[2]. Y aquí sostenemos que no hay que tener reparo (miedo, duda…) a formular este pensamiento, o principio, de una forma incluso más concreta y contundente: que ni la informática ni las telecomunicaciones se desarrollan, desde hace decenios, con capacidad para subvenir a necesidades imperiosas de las sociedades humanas, ni de las que consideramos subdesarrolladas o "primitivas", ni de las nuestras, a las que un tanto frívolamente consideramos desarrolladas y "evolucionadas".

Esta declaración (u observación, en un sentido mucho más exacto que la ley de Moore) debe contribuir siempre a reconciliarnos con nuestra propia profesión y –lo que es más importante aún– con nuestra responsabilidad social. Aunque el mundo de lo tecnológico se desenvuelva con escasa voluntad de proyectarse directamente en el ámbito de lo social, pese a la evidente influencia que ejercen las tecnologías electrónicas en la sociedad contemporánea, alguien se ha de plantear el análisis de las consecuencias sociales de, concretamente, la LM, toda vez que ésta puede representar un resumen y una "condensación" del papel de la electrónica y sus aplicaciones en la actualidad social y económica.

Mientras tanto, nuestros jóvenes titulados –de Informática, Telecomunicaciones y otras varias especialidades– experimentan en muchas ocasiones condiciones de trabajo que en estos inicios del siglo XXI son netamente peores que las de la segunda mitad del XX. Y todos nosotros debemos reconocer que dos éxitos tecnológicos que confirman la LM, como son el ordenador portátil y el teléfono móvil, tienden a someternos –y cuántas veces acaban consiguiéndolo– a una tensión laboral que se extiende, ni más ni menos, que a las 24 horas al día y a los 7 días a la semana, borrando eficazmente las diferencias entre el quehacer laboral y el tiempo privado. A ellos les corresponde, más que a nadie, preguntarse si todo esto corresponde a las promesas que se les ha ido haciendo y, sobre todo, si no podría haber sido de otro modo (¡Pues sí, desde luego, si se hubieran seguido otras prioridades políticas, socio-económicas e, incluso, científico-técnicas).

Y no olvidemos que los espectaculares cambios técnicos que tienen lugar en nuestras sociedades desde los años de 1960 son simultáneos –y en gran medida solidarios– con guerras devastadoras sin pausa, con el incremento de la miseria en regiones enteras del planeta y en millones de seres humanos, con la renovación incesante de regímenes políticos abyectos y con pérdidas constatables de derechos y libertades, precisamente, en las sociedades consideradas más desarrolladas y vanguardistas.

La técnica, efectivamente, no resuelve por sí misma ninguno de los problemas básicos de la humanidad: ¡No puede hacerlo! Los problemas de distribución y toma de decisiones tienen una componente social que cuando se olvida se tornan irresolubles.

escéptico Rousseau, con su *Discurso sobre las ciencias y las artes* (1750).

[1] Movimiento eminentemente cultural cuyo inicio se data en 1986 y en Roma, a consecuencia de una protesta popular frente a la americanización, concretamente la de la comida "rápida" (también llamada "comida basura").

[2] Virilio, *op. cit.*, p. 46.

REFERENCIAS

[1] G. Moore, "Cramming more components onto integrated circuits", *Electronics*, vol. 38, April 19, 1965.

[2] S. Rial, *Paul Virilio y los límites de la velocidad*, Madrid, España: Campo de Ideas, 2003

[3] I. Illich, *Energía y equidad*, Barcelona, España: Barral, 1973

[4] N. Burgi, "Salariés acrobates pour travail sans filet", *Le Monde Diplomatique*, nº 660, Mar. 2009.

[5] M. Castells, *La ciudad informacional*, Madrid, España: Alianza, 1995

[6] P. Virilio, *La bomba informática*, Madrid, España: Cátedra, 1999

[7] J. Ortega y Gasset, *Meditación de la técnica*, Madrid, España: Revista de Occidente, 1968.

BIBLIOGRAFÍA COMPLEMENTARIA

A. Mattelart, *Historia de la sociedad de la información*, Buenos Aires, Argentina: Paidós, 2007

M. Castells, *La Galaxia Internet*, Barcelona, España: Mondadori, 2001

C. Honoré, *Elogio de la lentitud*, Barcelona, España: RBA, 2006

P. Virilio, Ciudad pánico. El afuera empieza aquí, Buenos Aires, España: Libros del Zorzal, 2006

Calidad de experiencia en el acceso a Web sobre redes móviles HSDPA

E. Vázquez, M. Álvarez-Campana, J. Vinyes
Departamento de Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid
Ciudad Universitaria s/n – 28040 Madrid
{enrique, mac, vinyes}@dit.upm.es

***Resumen*- Gracias a la mejora de prestaciones proporcionada por la tecnología HSDPA (High Speed Downlink Packet Access), el acceso móvil a Internet ha experimentado un espectacular crecimiento en los últimos años, constituyéndose en alternativa a otras tecnologías de acceso de banda ancha, como el ADSL o el cable. HSDPA se basa en el empleo de un canal descendente compartido de alta velocidad, por lo que las prestaciones alcanzables individualmente son inversamente proporcionales al número de usuarios activos en la célula. En este artículo se evalúa la calidad de experiencia que percibe un usuario sobre un acceso compartido HSDPA en presencia de múltiples usuarios. El estudio se centra en tráfico Web e incluye resultados analíticos y de simulación. Los resultados obtenidos pueden servir de base para el dimensionado y control de admisión en redes HSDPA, de forma que se ofrezca a los usuarios un nivel de calidad de experiencia prefijado.**

***Palabras Clave*- HSDPA, acceso a Web, calidad de experiencia**

I. INTRODUCCIÓN

El despliegue inicial de las redes UMTS se basó en un conjunto inicial de especificaciones técnicas del 3GPP [1] (Third Generation Partnership Project), denominado Release 99. Debido a las limitaciones de esta primera versión de especificaciones, en los últimos años los operadores se han apresurado a introducir ciertas mejoras previstas para versiones posteriores. Se trata de las tecnologías HSPA [2] (High Speed Packet Access), término bajo el cual se agrupan un conjunto de extensiones al interfaz radio UMTS original encaminadas a mejorar las prestaciones tanto en sentido descendente, caso de HSDPA (High Speed Downlink Packet Access), como en sentido ascendente, mediante EUL (Enhanced Uplink).

Se puede considerar que con la implantación de las tecnologías HSPA se produce la entrada a los sistemas de comunicaciones móviles de "generación 3,5" (3,5G). La evolución de los sistemas 3GPP no termina ahí, y los operadores ya están preparando la introducción de nuevas extensiones como HSDPA Evolved (también conocida como HSDPA+) y, a más largo plazo, LTE (Long Term Evolution). Estas mejoras marcarán la entrada en las redes móviles de cuarta generación, barajándose caudales objetivo de decenas e incluso la centena de Mbit/s.

Las tecnologías HSDPA y EUL definidas respectivamente en la Release 5 y 6 del 3GPP, son ya realidad en la mayoría de las redes UMTS en funcionamiento. Con ellas se posibilita alcanzar tasas de pico teóricas de hasta 14,4 Mbit/s en bajada y 5,7 Mbit/s en subida, respectivamente.

HSDPA [3] permite ampliar el caudal de bajada para servicios modo paquete hasta 14,4 Mbit/s (teóricos) mediante la introducción de nuevas funcionalidades en el interfaz radio y en la red de acceso UMTS. Estas novedades traen consigo, además, una reducción de la latencia y un uso más eficiente del espectro. Se trata pues de una solución atractiva tanto para los usuarios como para los operadores.

En la práctica, las prestaciones alcanzables con HSDPA dependen de múltiples factores: configuración de la célula, condiciones de propagación, categoría del terminal, etc. No obstante, por proporcionar un valor orientativo, en la mayoría de las ocasiones es posible alcanzar velocidades de descarga en torno a 1 ó 2 Mbit/s.

Las nuevas características en las que se apoya HSDPA se pueden resumir en el empleo de un nuevo canal descendente compartido HS-DSCH (High Speed Downlink Shared Channel), junto con una serie de funcionalidades añadidas al Nodo-B que se agrupan bajo una nueva capa de control de acceso al medio MAC-hs. Dichas funciones incluyen un mecanismo de planificación rápida de paquetes junto con el empleo de modulación y codificación adaptativa (AMC) y un esquema de retransmisiones híbrido (H-ARQ).

En la Fig. 1 se ilustra la compartición dinámica del canal HS-DSCH entre los usuarios de HSDPA de una célula.

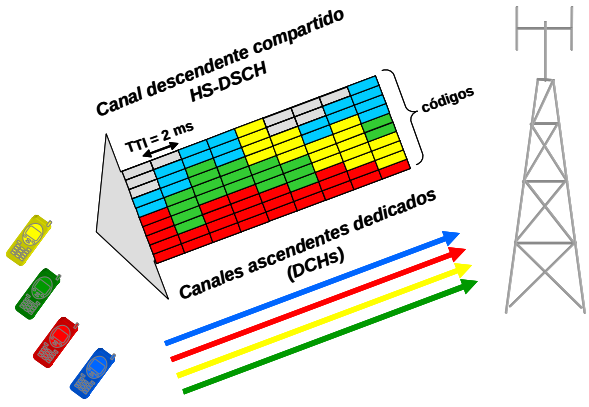


Fig. 1. Compartición de recursos radio en HSDPA

Para las comunicaciones en sentido ascendente, los terminales mantienen el empleo de canales dedicados DCH (Dedicated Channel).

El canal descendente compartido HS-DSCH es un nuevo canal de trasporte al que pueden asociarse uno o más canales físicos (códigos OVSF) con factor de expansión fijo de valor 16. Este valor permite la utilización de hasta 15 códigos para el canal HS-DSCH sobre una portadora WCDMA de 5 MHz. El operador puede dedicar una portadora completa a HSDPA o bien compartirla con servicios UMTS Release 99, siendo la segunda opción la más habitual en los despliegues iniciales de HSDPA. Este aspecto es importante ya que el número de códigos disponibles para HSDPA limita el caudal máximo de bajada en la célula.

El canal HS-DSCH se comparte dinámicamente entre los usuarios HSDPA de la célula bajo el control de un algoritmo de *scheduling* rápido gestionado en el Nodo-B. La asignación de recursos se efectúa con una granularidad de 2 ms, siendo éste el valor del parámetro TTI (Time Transmission Interval) empleado en el HS-DSCH. Dicho valor es cinco veces menor que el usado en UMTS Release 99.

A la hora de determinar las prestaciones de HSDPA, es necesario tener en cuenta la naturaleza compartida del canal HS-DSCH. Intuitivamente, esto implica que las prestaciones percibidas individualmente sean inversamente proporcionales al número de usuarios HSDPA simultáneamente activos en la célula. Este es precisamente el aspecto que se investiga en este artículo.

Para ello, se plantea como primer objetivo evaluar la capacidad dedicada equivalente (C_{eq}) que percibe un usuario en función de la capacidad total disponible en el canal descendente HSDPA y el número de usuarios que la comparten. C_{eq} es el valor de capacidad dedicada en exclusiva a un usuario que ofrecería a éste la misma calidad de experiencia que observa en el canal compartido HSDPA.

La aplicación objeto de este estudio es el acceso a páginas Web. En este caso, C_{eq} será el valor de capacidad de que dispone cada usuario para descargar sus páginas. Por tanto, C_{eq} determina el tiempo que debe esperar el usuario para que la página Web se muestre en su navegador y, por tanto, la calidad de experiencia que obtiene sobre HSDPA.

En la literatura se pueden encontrar estudios de prestaciones similares aplicados a otras tecnologías de acceso compartido, como son las redes de cable y GPRS [4][5]. Estos estudios evalúan la "tasa de circuito equivalente" que ofrecen dichas redes, parámetro similar a la capacidad dedicada equivalente utilizado aquí.

En el presente artículo se modela el comportamiento del canal compartido HSDPA para evaluar la capacidad equivalente percibida por los usuarios que acceden a páginas Web a través de dicho canal. Primero se presentan valores medios de capacidad equivalente estimados analíticamente. Luego se presentan resultados obtenidos mediante simulación correspondientes a la utilización del canal HSDPA, la capacidad equivalente percibida por los usuarios y el tiempo de descarga de páginas Web resultante.

Como paso previo, se hizo una revisión de estudios de tráfico Web publicados, a partir de la cual se decidió implementar en el simulador un modelo de generación de tráfico con distribuciones lognormales, ajustando sus parámetros según medidas publicadas recientemente.

El resto del artículo se organiza de la siguiente forma. En el apartado II se presenta el modelo analítico que permite estimar la capacidad equivalente a partir de los parámetros del sistema. En el apartado III se describe el modelo de generación tráfico Web considerado y los detalles del simulador. El apartado IV presenta los resultados más relevantes de las simulaciones, comparándolos con el modelo analítico. Por último, el apartado V resume las principales conclusiones del estudio y las líneas futuras de trabajo.

II. ANÁLISIS DE LA CAPACIDAD EQUIVALENTE

En esta sección se estudia analíticamente la relación que existe entre la capacidad dedicada equivalente (C_{eq}) introducida en el apartado anterior y los parámetros del sistema: la capacidad total disponible en el enlace HSDPA en sentido descendente (C), el número de usuarios que están compartiendo dicha capacidad (N) y las características del tráfico Web que genera cada usuario.

En el análisis se considera que cada usuario descarga páginas Web de longitud media L bits con un tiempo de lectura medio T_{off} segundos entre página y página. El valor de L será la suma del tamaño de la página HTML más el tamaño de todos los objetos, por ejemplo imágenes, incluidos en ella. El tiempo de lectura se cuenta desde que la página actual con todos sus objetos se ha descargado por completo hasta que se inicia la descarga de la página siguiente.

Si C_{eq} bit/s es la capacidad de que dispone un usuario cualquiera para descargar sus páginas, el tiempo medio que dura la descarga de una página es L/C_{eq} s. Por tanto, y según el comportamiento descrito, cada usuario genera L bits en un tiempo igual a $L/C_{eq} + T_{off}$ s. El tráfico total que generan los N usuarios dividido por la capacidad total C será el factor de utilización del enlace descendente (U). Por tanto:

$$U = \frac{N \cdot L/C}{T_{off} + L/C_{eq}} \qquad (1)$$

Sea α el factor de actividad que tendría un usuario que utilizara en exclusiva toda la capacidad disponible C. En este caso, el tiempo medio de actividad empleado en descargar una página sería L/C s y por tanto:

$$\alpha = \frac{L/C}{T_{off} + L/C} \qquad (2)$$

El factor α es un parámetro ideal, independiente de N, que caracteriza el comportamiento de un usuario aislado sobre un enlace de capacidad C. En la realidad la capacidad del enlace está compartida entre N usuarios, por lo que el factor de actividad real de cada uno de ellos será en general mayor que α. En efecto, a medida que el número de usuarios aumenta, la capacidad disponible para cada uno de ellos tiende a decrecer y, en consecuencia, el tiempo necesario para descargar cada página crece. Suponiendo que el tiempo de lectura medio Toff no varía, el aumento de tiempo de descarga de página a valores mayores que L/C hace que el factor de actividad real de cada usuario sea mayor que el valor dado por (2). Además, puesto que al aumentar N el enlace está más cargado y el tiempo de descarga de cada página es más largo, la cantidad de páginas que consulta cada usuario por unidad de tiempo se reduce.

De la ecuación (2) se deduce directamente:

$$T_{off} = \frac{L}{C} \cdot \frac{1-\alpha}{\alpha} \quad (3)$$

Sustituyendo esta expresión de T_{off} en la ecuación (1) y despejando C_{eq} /C resulta:

$$\frac{C_{eq}}{C} = \frac{\alpha \cdot U}{\alpha \cdot N - (1-\alpha) \cdot U} \quad (4)$$

La ecuación (4) permite calcular la capacidad equivalente C_{eq} que obtiene cada usuario, expresada como fracción de la capacidad total C, en función del número de usuarios N. El valor de α se calcula mediante (2) a partir de los parámetros L y T_{off} del modelo de tráfico Web utilizado. Sin embargo, para un número de usuarios N dado, es necesario calcular la utilización media resultante en el enlace, U, para poder obtener la fracción C_{eq} /C.

En [5] se estima U analíticamente mediante la siguiente expresión:

$$U = 1 - \left(\sum_{i=0}^{N} \left(\frac{\alpha}{1-\alpha} \right)^{i} \cdot \frac{N!}{(N-i)!} \right)^{-1} \quad (5)$$

Nótese que para un solo usuario, N=1, resulta U=α, es decir el factor de utilización del enlace coincide con el factor de actividad del usuario, como cabía esperar.

La Fig. 2 muestra un ejemplo de los resultados obtenidos mediante las expresiones anteriores para C=2000 kbit/s, L=125 kbytes y dos factores de actividad: α=0,01 y α=0,03. Estos valores corresponden a tiempos de lectura medios de 49,5 s y 16,2 s respectivamente, calculados a partir de los valores de α mediante la ecuación (2).

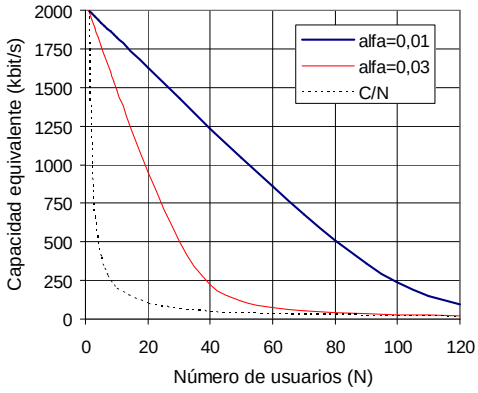


Fig. 2. Capacidad equivalente en función del número de usuarios para distintos valores del factor de actividad

Las curvas de la gráfica muestran claramente la ganancia que se obtienen al multiplexar N flujos de tráfico con un factor de actividad α pequeño. Cada uno de los N usuarios obtiene una capacidad equivalente C_{eq} significativamente mayor que C/N. Por ejemplo, para N=20 usuarios con α=0,03 resulta C_{eq} = 950 kbit/s aproximadamente. Es decir, cada usuario "ve" una capacidad de 950 kbit/s (en lugar de C/N = 2000/20 = 100 kbit/s) y, por tanto, percibe unas prestaciones comparables a las que tendría con un enlace dedicado de esa capacidad. El tiempo medio de descarga de página en este caso sería de 125 x 8 / 950 = 1,05 s, lo que resulta un valor satisfactorio. En cambio, con N=50 usuarios C_{eq} se reduce a 113 kbit/s aproximadamente y el tiempo de descarga medio es significativamente peor, 8,86 s, como muestra la Fig. 3.

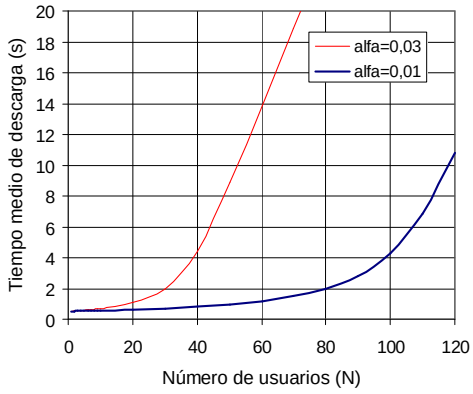


Fig. 3. Tiempo medio de descarga de página en función del número de usuarios para distintos valores del factor de actividad

Como se ha indicado antes, al aumentar N el tiempo de descarga de cada página es mayor y, por tanto, el número de páginas que consulta cada usuario por unidad de tiempo se reduce. Esto hace que la tasa media generada por cada usuario no sea constante, sino que decrezca al aumentar N, como muestra la Fig. 4. El descenso es más acusado cuanto mayor es el factor de actividad α, es decir, cuanto menor es el tiempo medio de lectura de página T_{off}.

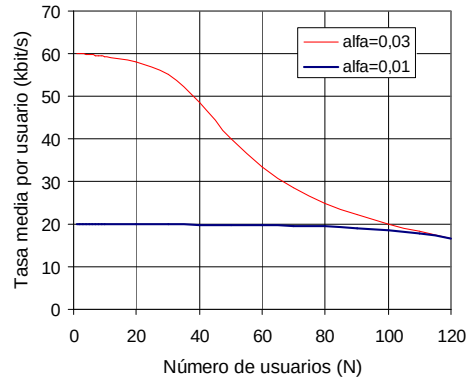


Fig. 4. Tasa media por usuario en función del número de usuarios para distintos valores del factor de actividad

A medida que la capacidad equivalente se reduce, los usuarios pasan más tiempo esperando a que las páginas se descarguen antes de poder leerlas. Por tanto, el porcentaje del tiempo que el usuario está esperando a recibir sus páginas en lugar de leyéndolas puede verse como otra medida de la calidad de experiencia. Este porcentaje de tiempo no es otra cosa que el factor de actividad *real* de cada usuario cuando hay N usuarios compartiendo el canal HSDPA. (Recuérdese que α es el factor de actividad *ideal* que tendría un usuario cuando N=1, es decir, cuando dispone de toda la capacidad del canal). Los resultados obtenidos se muestran en la Fig. 5. Para α=0,03 se observa que el tiempo dedicado a esperar por las páginas crece rápidamente a partir de N=30 usuarios.

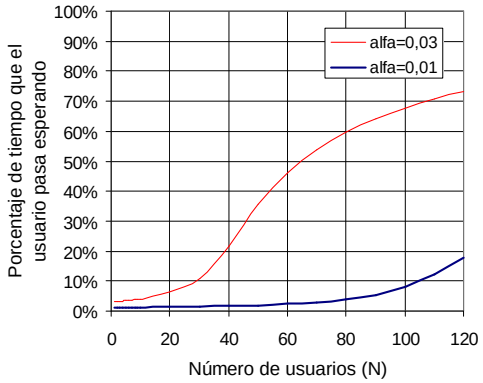


Fig. 5. Porcentaje de tiempo que un usuario pasa esperando por la descarga de sus páginas en función del número de usuarios para distintos valores del factor de actividad

En resumen, si se establecen objetivos de calidad ofrecida a los usuarios en términos de un límite inferior para la capacidad equivalente, o bien un límite superior para el tiempo medio de descarga, se puede calcular el máximo número de usuarios admisible para un valor de capacidad dado. Igualmente, si se toma como dato el número de usuarios a atender, puede dimensionarse la capacidad necesaria.

Sin embargo, las expresiones anteriores dan información sobre valores medios para el conjunto de los N usuarios, pero no sobre los valores de capacidad, retardo, etc. que efectivamente observa cada uno de ellos. En la práctica puede ser deseable establecer objetivos de calidad a cumplir para determinado porcentaje de las páginas consultadas o determinado porcentaje de los usuarios, por ejemplo que el 90% de los usuarios obtenga una capacidad equivalente superior a 128 kbit/s.

Para tener en cuenta este tipo de requisitos de calidad, en los apartados siguientes los valores de C_{eq} y de U en función de N se estiman mediante simulación. El simulador desarrollado permite estimar también los tiempos de descarga de página que observan los usuarios del servicio Web, los cuales son una métrica esencial para evaluar la calidad de experiencia percibida a medida que aumenta el número de usuarios.

En el apartado III se describe el modelo de tráfico Web utilizado y las características principales del simulador. En el apartado IV se muestran los resultados obtenidos. Por lo que respecta a valores medios se ha comprobado que las estimaciones del simulador coinciden con los resultados analíticos presentados antes. Después se presentan los resultados de simulación relativos a distribuciones de probabilidad y percentiles de las variables de interés.

III. Simulación de acceso a Web sobre HSDPA

A. *Modelo de tráfico Web*

En los últimos años se han realizado numerosos estudios que miden el tráfico generado por el acceso a páginas Web en diferentes escenarios y proponen distintas distribuciones estadísticas que se ajustan a los datos observados [6][7][8][9]. También se han publicado trabajos que resumen y comparan modelos propuestos por otros autores [10][11]. La diversidad de modelos puede explicarse por varias razones, principalmente las diferencias entre los escenarios donde se han tomado las medidas (ej. acceso por módem telefónico, acceso por cable o ADSL, acceso móvil) y la propia evolución del tráfico Web a lo largo de los años (ej. tamaño creciente de las páginas y contenido multimedia incluido en ellas [12] [13]).

Además de los factores anteriores, que afectan a las características de las páginas Web y el comportamiento de los usuarios que se quieren modelar, se observan diferencias en las medidas que se toman como punto de partida y los métodos aplicados para derivar el modelo a partir de ellas. Por ejemplo, en unos casos se parte de medidas de páginas consultadas en determinados servidores (por ej. [9]) mientras que en otros (ej. [10]) se mide el tráfico en un enlace. Algunos estudios [6] analizan el tráfico capturado a nivel de aplicación teniendo en cuenta detalles del protocolo HTTP, mientras que otros manejan medidas a niveles inferiores, a partir de las cuales hay que deducir qué paquetes corresponden a una misma página, por ejemplo mediante temporizadores [7][10].

En este artículo se utiliza un modelo con una estructura básica común a buena parte de los trabajos consultados. Esta estructura se basa en la descarga de una secuencia de páginas Web por parte de cada usuario, cada una de las cuales se compone de un objeto principal (HTML) y otros objetos adicionales (por ejemplo imágenes) que deben descargarse para poder mostrar la página completa. A continuación el usuario emplea cierto tiempo en leer la página mostrada antes de descargar la siguiente. Por tanto, cada usuario genera un patrón de tráfico que alterna periodos de actividad (descarga de una página) e inactividad (tiempo de lectura) a lo largo del tiempo.

En algunos estudios se modela también una escala de tiempo superior que corresponde al concepto de sesión, esto es, el periodo de tiempo durante el cual un usuario está descargando y leyendo páginas Web. Cuando la sesión termina el usuario deja de utilizar la aplicación hasta que comience la sesión siguiente. La escala de sesiones no es necesaria en este trabajo, ya que se quiere evaluar el número de usuarios que pueden compartir un acceso HSDPA simultáneamente manteniendo determinados niveles de calidad de servicio (o, en otras palabras, el número de sesiones Web simultáneas que se pueden soportar).

Las simulaciones que se detallan en el apartado IV se han realizado con un modelo de tráfico Web basado en [9] que usa distribuciones de probabilidad lognormales, pero ajustando los valores de algunos parámetros para incrementar el tamaño de las páginas descargadas en línea con medidas publicadas más recientemente [13]. Los valores de los parámetros de las distribuciones se muestran en la Tabla 1 junto con el resto de parámetros de la simulación.

B. *Implementación del simulador*

Con objeto de representar con detalle el efecto de la compartición del canal HS-DSCH entre los usuarios de una célula HSDPA, se ha desarrollado un simulador en lenguaje C, haciendo uso de la biblioteca de simulación ATLAS [14].

El programa simula un canal de capacidad C bit/s compartido entre N usuarios que generan tráfico Web de la forma descrita en el apartado anterior. En cada momento, la capacidad C se distribuye equitativamente entre los usuarios que se están descargando una página Web. De este modo,

cada vez que un usuario inicia una nueva descarga, la capacidad por usuario se recalcula adecuadamente, lo que implica retrasar los instantes de finalización de las descargas pendientes. El comportamiento descrito se ilustra mediante un ejemplo en la Fig. 6.

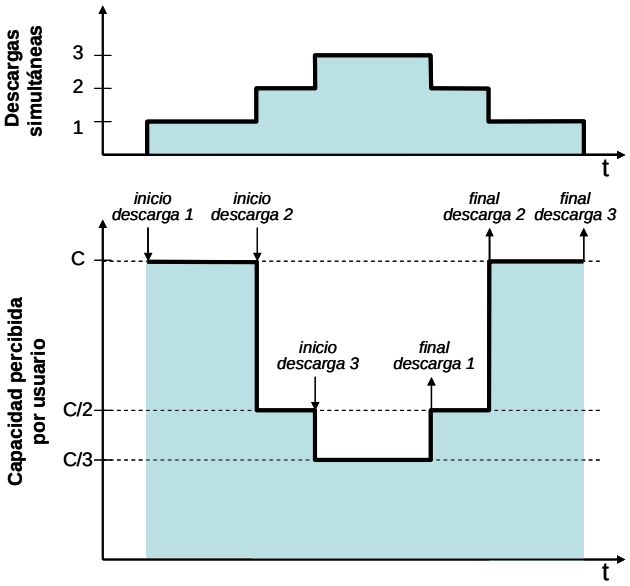


Fig. 6. Capacidad percibida según el número de descargas simultáneas

El simulador desarrollado calcula diversos estadísticos, entre los que se incluyen: utilización media del canal, número máximo de usuarios activos simultáneos, número total de páginas descargadas por usuario, tiempo medio de descarga de páginas y la capacidad equivalente percibida por el usuario. Los resultados obtenidos se refieren a los valores que permite alcanzar un canal compartido HSDPA sin incluir el efecto de los protocolos estándar utilizados en niveles superiores, tales como TCP y http. Para tenerlo en cuenta, el simulador descrito aquí se puede completar con modelos de simulación de los protocolos citados, disponibles en diversas herramientas de simulación de uso habitual.

IV. RESULTADOS

Los parámetros de simulación y los valores utilizados se resumen en la Tabla 1. Para la distribución lognormal se indican en cada caso los valores de la media (m) y la desviación estándar (s), a partir de los cuales se pueden calcular sus parámetros μ, σ.

Tabla 1. Parámetros de simulación

Nombre del parámetro		Valores simulados
C	Capacidad compartida (bit/s)	2000000
N	Número de usuarios, activos	10, 20, ... 120
Toff	Tiempo de lectura, (s)	Lognormal (m=40, s=300) / máx. 1000
Lp	Tamaño página HTML (kbyte)	Lognormal (m=25, s=50) / máx. 2000
Nobj	Número de objetos	Lognormal (m=8, s=30) / máx. 250
Lo	Tamaño objeto (kbyte)	Lognormal (m=12'5, s=120) / máx. 3000

Estos valores, basados como se ha dicho en modelos de tráfico Web consultados en la literatura, corresponden a un factor de actividad α=0,012, que es un valor intermedio entre los dos considerados en el apartado II.

La Fig. 7 muestra el porcentaje de utilización U de la capacidad del canal HSDPA en función del número de usuarios que lo comparten. La curva de línea continua muestra la estimación analítica de U según el procedimiento descrito en el apartado II. Los valores de U obtenidos mediante simulación, indicados con rombos, prácticamente coinciden con los del análisis. Esto da una prueba de que el funcionamiento del simulador es correcto.

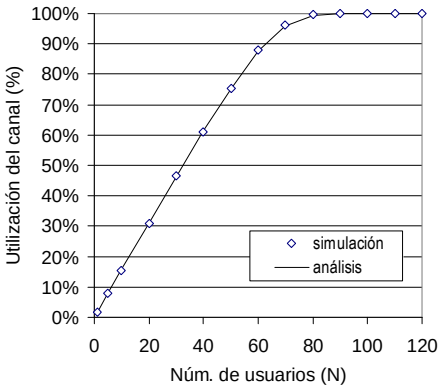


Fig. 7. Utilización del canal según el número de usuarios

Las gráficas de la Fig. 8 y la Fig. 9 muestran, respectivamente, la capacidad equivalente observada por cada usuario en función del número de usuarios y el tiempo medio de descarga de página correspondiente. Al igual que en el caso anterior, los resultados de simulación y los analíticos coinciden.

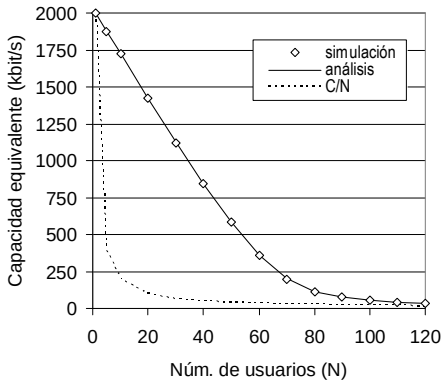


Fig. 8. Capacidad percibida según el número de usuarios

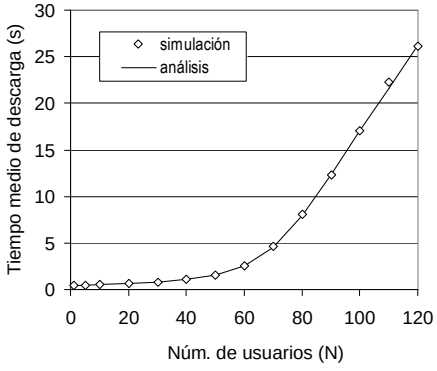


Fig. 9. Tiempo medio de descarga según el número de usuarios

Comparando en la Fig. 8 los valores de C_{eq} con la curva de trazos C/N (que correspondería a un canal dedicado de capacidad C/N para cada usuario) se observa la notable ganancia de multiplexión que permite el canal compartido de HSDPA para el tráfico Web. Con los parámetros de tráfico Web utilizados en la simulación, dicha ganancia alcanza un máximo cercano a 17 en el rango de 30 a 40 usuarios. Ver la Fig. 10.

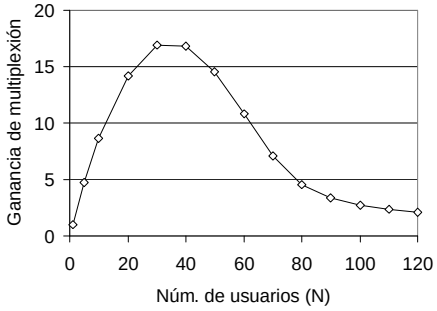


Fig. 10. Ganancia de múltiplexión (C_{eq}.N/C) en función de N

Para 60 usuarios, la ganancia es de 10,8, con una utilización del canal HSDPA de casi el 90% y unas prestaciones bastante buenas en cuanto a tiempo medio de descarga de página. A partir de 60 usuarios este tiempo crece rápidamente según indica la Fig. 9. Lógicamente, estos resultados dependen de los valores fijados para los parámetros del modelo de tráfico, en particular el tamaño de las páginas descargadas y el tiempo de lectura. Si estos valores cambian y el tráfico generado por usuario es distinto, el número de usuarios admisible deberá ser revisado.

Tras el estudio de valores medios, se muestran funciones de distribución obtenidas con el simulador. Estos resultados son útiles para estudiar el cumplimiento de requisitos de calidad que no se establecen en función de valores medios, por ejemplo garantizar que un porcentaje alto de los usuarios obtiene una capacidad igual o superior a un umbral dado, o bien que un porcentaje alto de las páginas se descargan en un tiempo no superior a un determinado umbral.

La Fig. 11 muestra la probabilidad de que la capacidad equivalente observada por un usuario cualquiera al descargar una página sea mayor que un valor dado. Se han simulado dos casos, representativos de condiciones de baja carga y de alta carga: 30 y 80 usuarios, respectivamente.

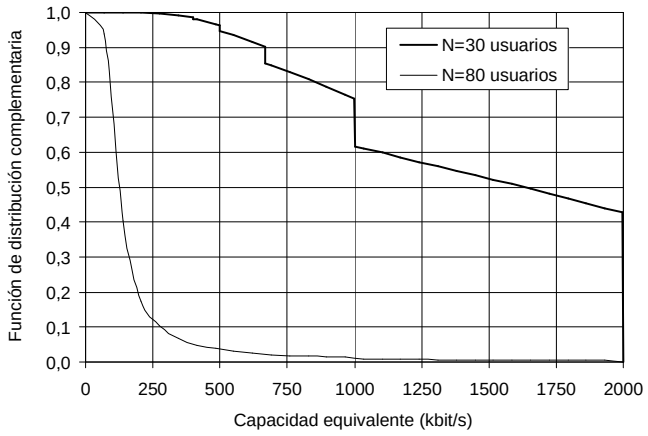


Fig. 11. Distribución de probabilidad de la capacidad equivalente

Cuando hay 30 usuarios, el canal HSDPA está poco cargado (su utilización es inferior al 50%). La probabilidad de que la capacidad observada sea 1 Mbit/s o superior está en torno a 0,75. La probabilidad de que la capacidad observada sea 2 Mbit/s, es decir de que un usuario descargue una página sin tener que compartir el canal con ningún otro usuario que esté activo simultáneamente, es un valor apreciable: 0,427.

Con 80 usuarios la situación es muy distinta. Ahora, la utilización del canal es cercana al 100% y la probabilidad de observar valores altos de capacidad es muy pequeña. Por ejemplo, la probabilidad de superar 300 kbit/s es inferior a 0,1 en este caso.

La Fig. 12 representa la probabilidad de que el tiempo necesario para descargar una página sea menor o igual que un valor dado para los mismos casos considerados antes: 30 y 80 usuarios. Comparando las curvas respectivas se observa el impacto del número de usuarios en el tiempo de descarga. Por ejemplo, con 30 usuarios más del 90% de las páginas se descargan en menos de 2 s. El porcentaje de páginas que tardan 10 s o más es inferior al 1%. En cambio, con 80 usuarios solo el 48% de las páginas se descargan en menos de 2 s, mientras que un 16% tardan más de 10 s. Casi el 9% de las páginas tardan por encima de 20 s.

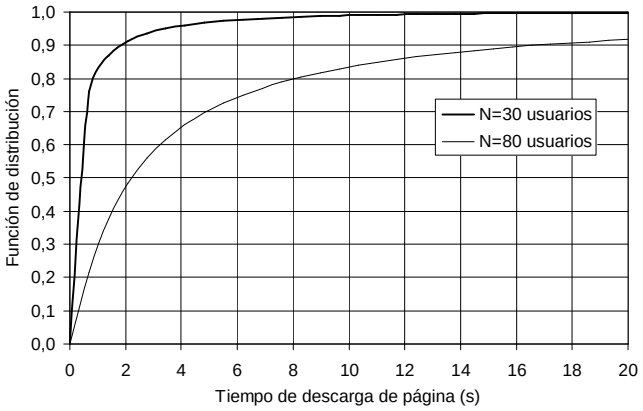


Fig. 12. Distribución de probabilidad del tiempo de descarga de página

Por último, las gráficas siguientes muestran percentiles para las dos variables, capacidad equivalente y tiempo de descarga de página, en función del número de usuarios simulado. En estas gráficas se incluyen también las curvas que indican los valores medios de dichas variables.

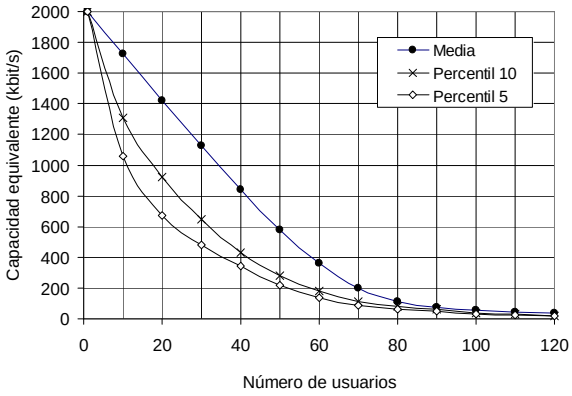


Fig. 13. Capacidad equivalente: media y percentiles

Según la Fig. 13, si se quiere que la capacidad equivalente media ofrecida a los usuarios sea, por ejemplo, 600 kbit/s se podrían admitir hasta 50 usuarios. Sin embargo, si lo que se quiere es garantizar que el 95% de las páginas se descarguen como mínimo a 600 kbit/s, el número de usuarios admisible se reduce a menos de la mitad.

El mismo tipo de requisitos se puede plantear en términos del tiempo de descarga. Según la Fig. 14, si se quiere que el 95% de las páginas tarden menos de, por ejemplo, 5 segundos, se pueden admitir 40 usuarios.

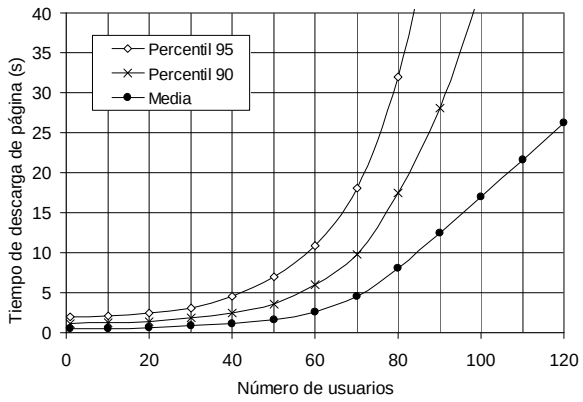


Fig. 14. Tiempo de descarga de página: media y percentiles

V. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se ha evaluado la calidad de experiencia de los usuarios Web que comparten el canal descendente de una célula HSDPA. Para ello, se ha presentado una estimación analítica de las prestaciones medias percibidas por cada usuario y, posteriormente, se ha desarrollado un simulador que permite estimar las funciones de distribución de probabilidad de las variables de interés. El simulador incluye un generador de páginas Web cuyos parámetros se han ajustado a partir de medidas de tráfico Web publicadas recientemente.

El estudio de prestaciones efectuado permite abordar el dimensionado de redes HSDPA en función de parámetros de prestaciones básicos como son la capacidad dedicada equivalente que obtiene cada usuario del canal compartido y el tiempo de descarga de página que observa.

Los resultados obtenidos muestran que un canal HSDPA puede ser compartido por varias decenas de usuarios Web con una calidad percibida satisfactoria, debido a que el factor de actividad de esta aplicación (esto es, el tiempo que el usuario está generando tráfico al descargar páginas Web frente al tiempo que emplea en leer dichas páginas) es bastante bajo.

Como trabajo futuro, se considera la extensión del estudio a otros tipos de tráfico (P2P, voz, vídeo, etc.) que tienen patrones de actividad diferentes. De este modo, se pretende desarrollar una herramienta de dimensionado flexible que pueda ajustarse a las mezclas de tráfico previstas en las redes HSDPA.

AGRADECIMIENTOS

Este trabajo ha sido financiado parcialmente por el proyecto CASERTEL-NGN (TSI2005-07306-C02-01).

REFERENCIAS

[1] Especificaciones de UMTS, servidor web del 3GPP (Third Generation Partnership Project), www.3gpp.org.

[2] UMTS Forum, “HSPA: High Speed Wireless Broadband, From HSDPA to HSUPA and Beyond”, junio 2005.

[3] 3GPP, “High Speed Downlink Packet Access (HSDPA); Overall description; Stage 2”, TS 25.308, marzo 2002.

[4] N.K. Shankaranarayanan, Z. Jiang, P. Mishra, “User-perceived performance of Web-browsing and interactive data in HFC cable access networks”, IEEE International Conference on Communications, ICC, Helsinki, junio 2001.

[5] N.K. Shankaranarayanan, Z. Jiang, P. Mishra, “Performance of a Shared Packet Wireless Network with Interactive Data Users", Mobile Networks and Applications, vol. 8, junio 2003.

[6] H. Choi, J. Limb, “A Behavioral Model of Web Traffic”, International Conference on Network Protocols, ICNP, Toronto, octubre 1999.

[7] C. Zhu, Y. Wang, Y. Zhang, W. Wu, “Different Behavioral Characteristics of Web Traffic between Wireless and Wire IP Network”, International Conference on Communication Technology, ICCT, Beijing, abril 2003.

[8] E. Casilari, J.M. Cano-García, F.J. González-Cañete, F. Sandoval, "Modelling of individual and aggregate Web traffic", 7th IEEE International Conference on High Speed Networks and Multimedia Communications, HSNMC, Toulouse, julio 2004.

[9] J.J. Lee, M. Gupta, “A new traffic model for current user web browsing behavior”, Intel Corp, 2007. http://blogs.intel.com/research/2007/09/a_new_traffic_model_for_curren.php

[10] E. Casilari, A. Reyes-Lecuona, F.J. González-Cañete, A. Díaz-Estrella, F. Sandoval, “Characterisation of Web Traffic”. IEEE GLOBECOM 2001, San Antonio, Texas, noviembre 2001.

[11] P. Tran-Gia, D. Staehle, K. Leibnitz, “Source Traffic Modeling of Wireless Applications”, International Journal of Electronics and Communications, vol. 55, num. 1, enero 2001.

[12] F. Hernández-Campos, K. Jeffay, F. Donelson Smith, “Tracking the Evolution of Web Traffic: 1995-2003”, 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems, MASCOTS, Orlando, Florida, octubre 2003.

[13] R. Levering, M. Cutler, “The portrait of a common HTML web page”, ACM Symposium on Document Engineering, Amsterdam, octubre 2006.

[14] M. Rümekasten, “ATLAS Reference Manual”, Department of Mathematics and Computer Science, University of Paderborn, 1991.

Plataforma Telemática de Integración de Estándares *End-to-End* para Salud Personal

I. Martínez[1], J. Escayola[1], J.D.Trigo[1], M. Martínez-Espronceda[2], L. Serrano[2], P. Muñoz[1], J. García[1]

[1] Aragon Institute for Engineering Research (I3A) / University of Zaragoza (UZ)
c/ María de Luna, 3, 50018 Zaragoza, Spain {imr, jescayola, jtrigo, pmugnoz, jogarmo}@unizar.es
[2] Electrical Electronics Engineering Dept. / Public Univ. Navarra (UPNA)
Campus de Arrosadía s/n. 31006 Pamplona, Spain {miguel.martinezdeespronceda, lserrano}@unavarra

***Resumen*— En los últimos años, se ha demostrado la necesidad de desarrollar estándares abiertos y componentes *middleware* que permitan la integración transparente e interoperable entre los diferentes elementos incluidos en una plataforma telemática. Este artículo aborda esta problemática, aplicada a los entornos de salud personal (p-Salud), presentando una propuesta basada en la integración de estándares extremo a extremo. Este concepto de estandarización *end-to-end* se destaca como el camino a seguir para la definitiva implementación de soluciones de p-Salud. Para ello, se analizan las líneas de diseño conformes a las más recientes evoluciones de los dos estándares europeos de referencia en este campo: ISO/IEEE11073 para interoperabilidad de dispositivos médicos actualmente orientado a entornos ubicuos y dispositivos llevables (*Personal Health Devices*, X73PHD), y EN13606 para el intercambio interoperable de Historia Clínica Electrónica (HCE). Por último, se presentan los resultados obtenidos en la implantación de la plataforma sobre dispositivos *wireless* y se analizan los retos pendientes en el diseño de un nuevo protocolo *end-to-end* para la armonización de ambos estándares (*End-to-End Standard Harmonization Protocol*, E2ESHP) que permita la transferencia de la solución propuesta al sistema de salud.**

***Palabras clave*— EN13606, estándares extremo-a-extremo, ISO/IEEE11073, plataforma telemática, salud personal.**

I. Introducción

A lo largo de los últimos años se han incorporado importantes avances al ámbito de las aplicaciones orientadas a la gestión de la salud gracias al desarrollo de las tecnologías de la información y las comunicaciones. El entorno de aplicación de estos servicios de telemedicina ha pasado de estar localizado principalmente en el ámbito local del hospital o las unidades de cuidados intensivos, a extenderse al entorno del paciente como centro del servicio de salud. Esta orientación tiene como consecuencia la creación de nuevos conceptos telemáticos como las redes domiciliarias, personales y corporales (*Home/Personal/Body Area Networks*, HAN/PAN/BAN). Esta evolución proporciona al paciente/usuario la posibilidad de desplazarse o cambiar de localización mientras su salud sigue controlada o monitorizada dando lugar, a su vez, a la evolución de los servicios de telemedicina (tradicionalmente considerados como de e-Salud) hacia entornos de aplicación inalámbricos o móviles (m-Salud), alcanzando soluciones ubicuas independientes de la localización de la aplicación (u-Salud) y llegando, como se presenta en este artículo, a aplicaciones personales centradas en el paciente/usuario (p-Salud).

Todos estos escenarios de uso están caracterizados por los dispositivos médicos (*Medical Devices*, MDs) que incorporan. Estos MDs están equipados con sensores específicos para obtener las señales biológicas del paciente/usuario (señal electrocardiográfica (ECG), presión sanguínea, pulso, peso, temperatura, etc.), y posteriormente evaluadas por el propio usuario o por personal especializado. Los MDs también han evolucionado hacia entornos de p-Salud convirtiéndose en dispositivos de salud personal (*Personal Health Devices*, PHDs) que van más allá de la tarea de adquisición/envío de la señal ya que, además de permitir obtener medidas en cualquier lugar, hacen uso de tecnologías inalámbricas y dispositivos portátiles para enviar la información a sistemas remotos.

Tal grado de libertad a la hora de elaborar dispositivos con sensores de alta fiabilidad, combinados con un abanico de posibilidades orientadas al paciente y su uso personal, ha incrementado el número de equipos disponibles en el mercado. El problema de la interoperabilidad, ya presente en los MDs de primera generación se agrava todavía más ahora con las nuevas aplicaciones incorporadas. Las soluciones de p-Salud basadas únicamente en el tipo de información obtenida de los sensores, funcionan en numerosas ocasiones con un cierto tipo de dispositivo (seleccionado durante la etapa de diseño de la aplicación) y son incompatibles con equipos de similares características, o incluso más apropiadas al entorno. Así mismo la sustitución del equipo se hace más complicada en caso de avería al depender de un modelo concreto. Es necesario, pues, hacer un esfuerzo por lograr la interoperabilidad y los propios fabricantes comienzan a darse cuenta del potencial tecnológico y económico de incorporar estándares médicos en sus MDs [1].

En este contexto, varias organizaciones han estado persiguiendo este objetivo durante años. El Comité Europeo de Estandarización (CEN) junto con el Comité Técnico CEN/TC251 son las principales instituciones europeas; y a nivel nacional, la Asociación Española de NORmalización (AENOR) y su Comité Técnico AEN/CTN139, grupos en los cuales nuestro equipo ha estado colaborando activamente [2]. Existen varias normas y estándares médicos aplicables a la interoperabilidad en sus diferentes especializaciones: DICOM para imágenes médicas [3], SCP-ECG para intercambio de señales ECG [4], HL7 para intercambio de mensajes de ámbito médico [5], ISO/IEEE11073 para interoperabilidad de MDs [6], y EN13606 para almacenamiento e intercambio del Historial Clínico Electrónico (HCE) del paciente [7].

La incorporación de todos estos estándares existentes en un mismo sistema es complicado y requiere de un gran esfuerzo de integración. Para ello, es imprescindible la coordinación entre instituciones, empresas y otras organizaciones tanto sanitarias como de investigación. Ante este panorama surgen otros dos organismos: *Integrating the Healthcare Enterprise* (IHE) [8] que trata de buscar, junto con los fabricantes de MDs, la mejor solución para cada servicio específico; y *Continua Health Alliance* [9], formada por 22 compañías del sector de tecnologías sanitarias, que persigue la incorporación de tecnologías interoperables en los dispositivos así como promover el uso de estos sistemas en las aplicaciones tanto a nivel profesional como cotidiano, planteando como reto la obtención de un certificado de normalización en forma de logotipo a incluir en los productos comerciales compatibles con los correspondientes estándares.

Fruto de este trabajo y de entre los normas que han sido adoptadas y aprobadas, para el contexto planteado en este desarrollo destacan dos: EN13606 e ISO/IEEE11073 (X73). Estos estándares, como se estudiará en este artículo, han evolucionado considerablemente en los últimos años: EN13606 ha pasado de ser un pre-estándar (ENV13606) a aprobarse recientemente completando sus partes 4 y 5; y X73, cuyas primeras versiones se orientaron a las comunicaciones en el Punto de Cuidado (X73PoC) del paciente [10], se ha optimizado hacia su versión actual que incluye las tecnologías de transmisión emergentes (como USB, Bluetooth, WiFi o ZigBee, no contempladas inicialmente) y se orienta a entornos de p-Salud sobre dispositivos llevables (X73PHD) [11].

Este artículo estudia la problemática de interoperabilidad y la estandarización, aplicada a entornos de p-Salud, presentando una plataforma telemática de integración de estándares. En la Sección II se estudia la evolución del diseño y las migraciones de la plataforma, conforme a las versiones de X73 y EN13606. En la Sección III se describe la arquitectura de la plataforma detallando sus características técnicas cumpliendo la evolución de X73, su integración con EN13606 y la inclusión de un nuevo protocolo de armonización de estándares (*End-to-End Standard Harmonization Protocol*, E2ESHP). La Sección IV analiza el diseño basado en estándares y detalla su aplicación en la implementación, cumpliendo los requisitos específicos conforme a ambos estándares X73 y EN13606. La Sección V presenta los resultados de la implementación, discute sus puntos abiertos, y analiza los nuevos retos de implantación en micro-controladores e incluyendo señales a tiempo-real para conformar una solución transferible al sistema de salud.

II. Evolución de la plataforma

Las evoluciones sufridas en los últimos años por los estándares X73 y EN13606 y sus implicaciones, tanto en la arquitectura funcional del sistema como en las reglas de diseño del modelo de comunicaciones y sus protocolos, han requerido varias migraciones de implementación de la plataforma que se detallan en las siguientes fases (ver Fig.1):

- Fase 1 (*plataforma1.0-αlfa*) [12]. Primera solución *end-to-end* dividida en dos subsistemas: el subsistema de adquisición que permite la conexión (vía RS-232 e IrDA) entre los dispositivos médicos y un elemento central (*gateway*) conforme a X73PoC, y el subsistema de almacenamiento que soporta el intercambio de la información médica en un servidor de HCE conforme a ENV13606. Se basó en lenguajes C/C++ y Java, y operaba sobre SO Linux (permitiendo simular la consola Linux en computadoras con SO Windows mediante CYGWIN-POSIX/ GNU.GCC). La arquitectura X73PoC se basó en elementos de servicio ACSE, ROSE y CMISE definidos en librerías ASN1.C (ASNX en X73) mediante sintaxis abstracta (MDDL) y sintaxis de transferencia (MDER, BER, y PER).
- Fase 2 (*plataforma1.5-βeta*) [13]. Segunda solución *end-to-end* que mantiene los dos subsistemas, pero evoluciona de X73PoC a X73PHD: independizando la capa de transporte (mediante un *handler* compatible con TCP/IP sobre USB y Bluetooth), incluyendo envío de datos episódico (al anterior periódico) mediante un sistema de *buffers* para las PDUs de cada capa de la pila, y optimizando el *gateway* hacia el concepto de *Compute Engine* (CE) sobre una nueva máquina de estados finita. La arquitectura integra tanto X73PoC como X73PHD, manteniendo las estructuras ACSE, ROSE y CMISE, pero flexibilizando la sintaxis MDER/BER/PER.
- Fase 3 (*plataforma2.0-release*). Evolución actual que integra los dos subsistemas mediante un nuevo protocolo extremo a extremo, incluye todas las evoluciones tanto de X73PHD como de EN13606, incorpora un nuevo GUI y, sobre todo, migra la implementación de MDs y CEs hacia dispositivos móviles (*PDAs*, *SmartPhones*). La arquitectura completa se basa en X73PHD, se optimiza el código C++/Java sobre Windows, y se implementa la comunicación Bluetooth optimizando las funcionalidades ubicuas (*plug-and-play* y *hot-swap*) para permitir su aplicación a soluciones de p-Salud.

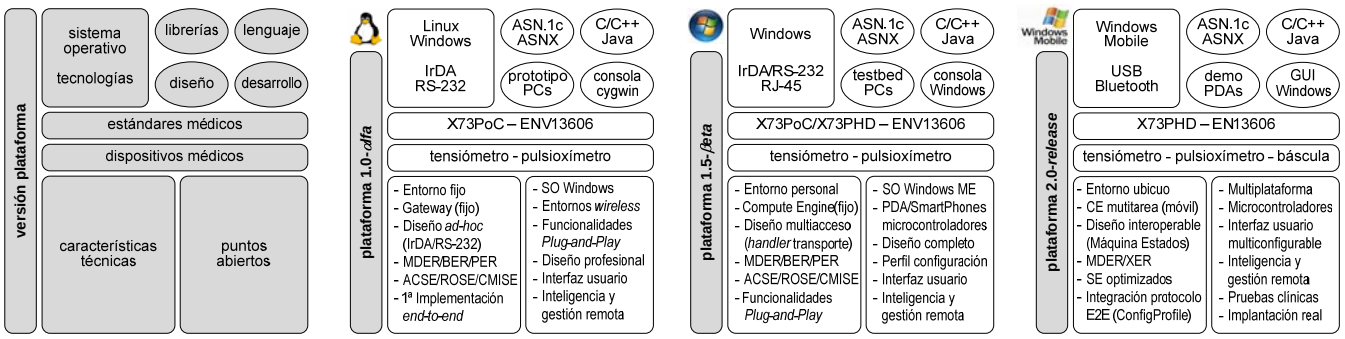


Fig. 1. Esquema de evolución de migraciones de la plataforma telemática de salud personal

III. ARQUITECTURA DE LA PLATAFORMA

Uno de los grandes retos en la línea de investigación de desarrollo e integración de estándares es su implementación en soluciones reales de p-Salud, transferibles al sistema sanitario. En los últimos años se han desarrollado contribuciones para el estudio de la viabilidad de aplicación de X73 en entornos sanitarios implementando soluciones para monitorización de pacientes centradas en el PoC [14]-[16], nuevas propuestas para PHDs para testear la evolución del estándar [17]-[18], o implementaciones aisladas de EN13606 en sistemas de salud [19]. Sin embargo, no hay antecedentes europeos de soluciones estándares extremo a extremo basadas en X73PHD y EN13606 y orientadas a p-Salud, como se presenta en este artículo.

La arquitectura de la plataforma (ver Fig. 2) está basada en una pasarela (CE), que concentra toda la información recogida por los dispositivos (MDs) de diferentes pacientes que definen un entorno ubicuo y de salud personal del paciente. Este CE se comunica a través de la red externa de acceso y transporte, con un servidor de monitorización que gestiona distintos CEs y reúne toda la información proveniente de cada escenario para actualizar la HCE del paciente. Las características de cada elemento que conforma la arquitectura del sistema son:

- *MDs*. La adquisición de datos médicos sigue un formato propietario (aunque incluyen interfaces universales, USB o Bluetooth, los protocolos que emplean son propietarios). Así, estos adaptadores crean la especialización para el MD que genera su modelo de información específico y establece la máquina de estados finitos (*Finite State Machine*, FSM) permitiendo así a los MDs no compatibles X73PHD actuar como agentes nativos de una comunicación X73PHD.
- *CEs*. El dispositivo de pasarela está diseñado como un manager nativo X73PHD que recoge toda la información médica proveniente de MDs y emula su FSM. La información es almacenada en un fichero de datos X73PHD que, junto con el perfil de configuración específico (*Config Profile*), sirve como entrada de datos para el proceso de creación de tramas para el protocolo E2ESHP. E2ESHP es una propuesta para un nuevo protocolo diseñado para la integración *end-to-end* con el tipo de datos definido en la HCE conforme a EN13606. Este protocolo está en fase de desarrollo pero su diseño permite integrar la información adquirida por los MDs en la HCE de forma transparente y posibilita, a partir de una consulta en la base de datos o de modificaciones del *Config Profile* del correspondiente caso de uso, administrar y gestionar remotamente los CEs (gestor de toma de medidas, envío de avisos, alarmas, etc.) y las especificaciones de cada MD (umbrales de funcionamiento, identificadores de paciente, etc.).
- *MS*. Está compuesto por dos entidades. La primera actúa como servidor E2ESHP puesto que se encarga de recibir los datos de X73PHD, decodificando tramas E2ESHP y extrayendo los datos X73PHD apropiados (clasificando información por usuario asociado) para almacenarlos en la base de datos. El segundo implementa una doble función cliente/servidor EN13606 aceptando peticiones EN13606 de información almacenada en la base de datos, y generando extractos EN13606 siguiendo sus arquetipos.

Con la arquitectura de plataforma propuesta, las guías de diseño y el proceso de implementación deben garantizar algunas especificaciones técnicas referentes a los estándares X73PHD y EN13606 como se detalla en la sección siguiente.

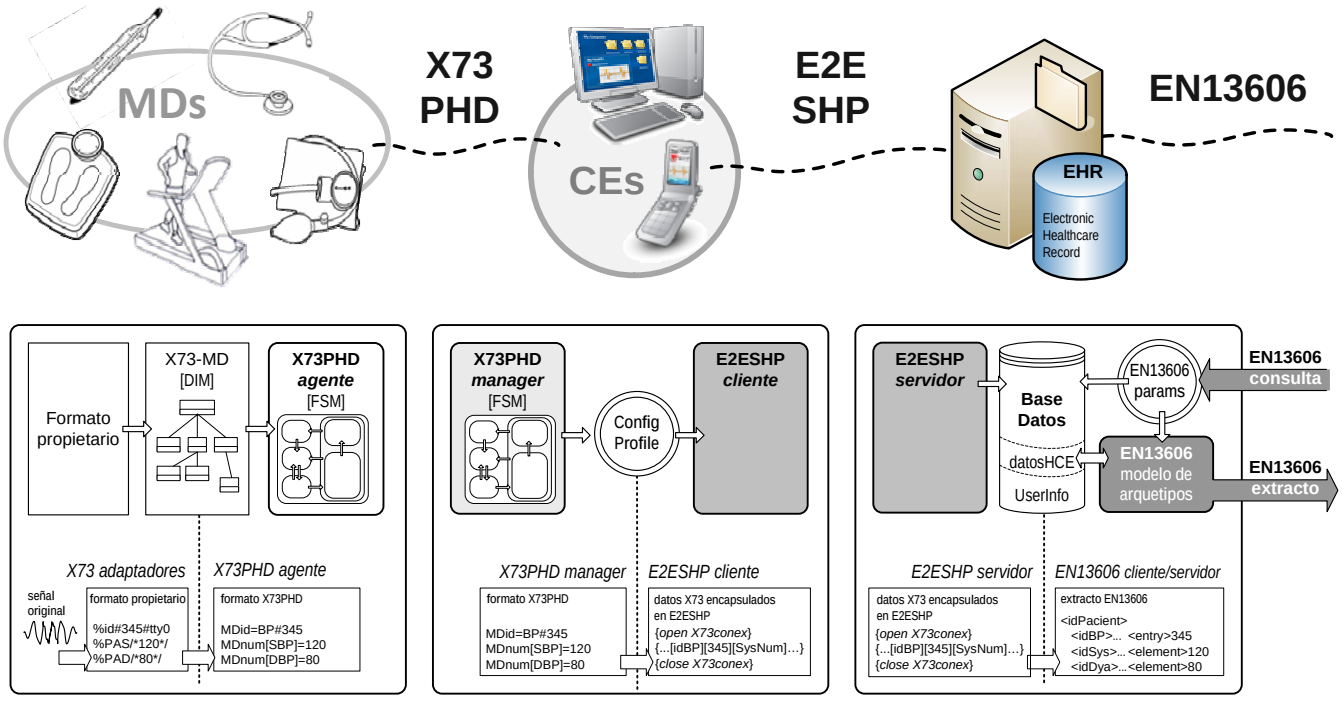


Fig. 2. Arquitectura de la plataforma telemática para integración de estándares extremo-a-extremo

IV. DISEÑO BASADO EN ESTÁNDARES APLICADO A LA IMPLEMENTACIÓN DE LA PLATAFORMA

A. Requisitos conforme al estándar X73PHD

Las soluciones de p-Salud implementadas sobre el estándar X73PHD se basan en redes con topología en estrella compuesta por varios MDs (agentes) y un CE (manager). El proceso básico de comunicación se fundamenta en las peticiones de asociación (*association requets*) que el manager recibe del agente, sobre las que se decide si aceptarlas (y, consecuentemente, entrar en el estado de operación) o rechazarlas. A partir de la nueva evolución del protocolo X73PHD, hay varios puntos clave a tener en cuenta en el proceso de implementación de una aplicación completa. Algunos de los más representativos se analizan a continuación:

- *Adaptación de MDs.* Actualmente, existe en el mercado una carencia de disponibilidad de dispositivos compatibles con X73PHD. Incluso los que incorporan en sus especificaciones interfaces universales (como USB o Bluetooth) funcionan con formatos propietarios de adquisición y envío, por lo que la plataforma mantiene el uso de adaptadores X73 para compatibilizar estos MDs a la espera de equipos comerciales.
- *Integración de MDs.* La nueva plataforma propuesta tiene capacidad de incorporación de un número de dispositivos mucho mayor que en la plataforma anterior haciendo uso de una combinación de tecnologías de transporte tanto cableadas (USB o RJ-45) como inalámbricas (Bluetooth, WiFi o Zigbee). Además, X73PHD define un conjunto nuevo de especificaciones que facilita esta tarea independizando el diseño de MDs de sus respectivas capas de transporte.
- *Mobilidad de CEs.* El nuevo concepto de plataforma ubicua permite el diseño de un manager con posibilidad de ser ejecutado sobre dispositivos inalámbricos (teléfonos móviles, *PDAs*, *Smartphones*). Este avance es clave en el diseño de los nuevos casos de uso que se estudian en el grupo de trabajo PHDWG orientados a p-Salud: autocontrol de la salud, entornos *fitness*, etc. Sin embargo, su diseño implica ciertas consideraciones hardware/software: autonomía de las baterías, complejidad de uso, estabilidad del dispositivo, diseño centrado en usuario, optimización del código, etc.
- *Optimización de CEs.* Con las nuevas reglas de diseño X73PHD es necesario analizar la eficiencia del protocolo y monitorizar su carga de tráfico para incorporar las pertinentes modificaciones en las reglas de codificación MDER y en las definiciones de tipos de datos ASN.1 para optimizar el intercambio de información en términos de tiempo y coste.
- *Comunicación MD-CE.* El proceso de comunicación punto a punto, junto con el diseño de la máquina de estados FSM es un aspecto clave en la implementación dado que la pila de protocolos completa debe diseñarse en MD y en CE. A partir del demostrador básico que incluía la plataforma anterior, se ha de incorporar un nuevo gestor de tramas e intercambio de PDUs y optimizar el diseño de un interfaz gráfico (GUI) multimodal tanto para la monitorización de dispositivos y muestras del usuario como para la gestión del protocolo a nivel de depuración y evaluación, que permitan convertir la plataforma en un entorno completo de pruebas a usar como herramienta de verificación de nuevas implementaciones X73PHD, chequeo de estabilidad de las distintas versiones del protocolo, análisis de carga de tráfico, retardos, etc.

A partir de estas consideraciones, para el diseño de la nueva plataforma hay que tener en cuenta que la norma X73PHD va orientada a dispositivos de uso personal que, como se ha comentado, poseen unas características muy restrictivas en cuanto a capacidad de procesado (velocidad y memoria) y autonomía. Uno de los desafíos de la plataforma es conseguir programar el agente X73PHD en un sistema basado en microcontrolador. La reducción de complejidad a conseguir, teniendo como referencia la plataforma anterior basada en X73PoC, es drástica al mismo tiempo que se ha de optimizar el uso de memoria. Además, hay que independizar el diseño de la tecnología de transporte empleada teniendo en cuenta las recomendaciones de PHDWG: el tipo de señal a transmitir queda caracterizado por la estructura de los datos (muestras simples o vectores), su tamaño, y la frecuencia de transmisión requerida; lo que puede implicar una gestión más eficiente en unas tecnologías que en otras (proceso de asociación y autenticación, debido al tamaño de cabeceras y velocidades de transmisión). Por otro lado, es determinante el entorno de aplicación del dispositivo dado que, el uso de tecnologías inalámbricas puede estar desaconsejado en algunos escenarios médicos por poder interferir con otros equipos electrónicos. Al mismo tiempo, la distancia al manager y su ubicación requerirán enlaces cableados o inalámbricos (con diferente tecnología en cada caso), pero manteniendo la homogeneidad del diseño X73. Por último, se ha de tener en cuenta su posible transferencia al sistema sanitario debido a que para cada tecnología se precisa implementar los módulos básicos de interfaz físico (radio o cable), y el controlador de protocolo que gobierna la transmisión y las comunicaciones con el resto de dispositivos; por lo que cuestiones como precio, disponibilidad, consumo, tamaño y complejidad de desarrollo sobre el sistema global, son aspectos claves a analizar en el diseño propuesto.

Con todo ello, para el nuevo desarrollo de *plataforma2.0-release* se ha hecho uso del mismo lenguaje de programación que en las dos plataformas previas: C++. Algunos de los criterios que se han tenido en cuenta para su selección son:

- Necesidad de uso de punteros para la gestión de árboles de objetos, gestión eficiente de memoria y tramas de datos.
- Experiencia de desarrollo con el lenguaje C/C++.
- Acceso a bajo nivel *hardware* y posibilidad de desarrollo en sistemas empotrados y migración a microcontroladores.
- Integración con entornos de diseño de aplicaciones de ventanas basados en *Microsoft Foundation Class* (MFC).

El entorno de desarrollo ha sido básicamente Microsoft Visual Studio C++ y su versión para sistemas empotrados Microsoft eMbedded Visual C++. Todo el código propio del protocolo X73PHD se ha definido sobre tipos de datos ASN.1, sintaxis MDER optimizada y se ha desarrollado con C++ para la implementación de la arquitectura en tres niveles (Modelo de Información (DIM), de Servicios, y de Comunicaciones), y librerías MFC para los interfaces visuales. Estas librerías permiten encapsular tanto el código perteneciente al protocolo como las funcionalidades requeridas en clases más sencillas. Además proporciona las herramientas necesarias para la creación de aplicaciones tanto de ventana como de consola de una manera sencilla tanto para plataforma Win32 como sistemas empotrados basados en Windows CE.

Para la implementación del estándar se ha seguido el borrador de la norma *ISO/IEEE P11073-20601/D20 Draft Standard for Health informatics - Personal health device communication - Application profile - Optimized exchange protocol* en su versión 20 lanzada en mayo del 2008 [11]. Esta versión ha sufrido muy pocos cambios desde entonces, estando localizados en aquellos apartados relacionados con determinadas especializaciones de dispositivos para incrementar su compatibilidad. Dado que son muchas las características que ofrece el protocolo y algunas de ellas no tienen sentido dependiendo del tipo de dispositivo que se emplea en el sistema, es importante hacer una selección de dichas propiedades. Aún así, mientras que en la plataforma anterior se buscaba poder desarrollar un demostrador que contuviera la mínima parte de las características del X73PoC para poder funcionar con una configuración determinada, en esta ocasión el sistema ha de ser capaz de reconocer la mayor parte de los mensajes y la máquina de estados por completo. Entre las nuevas características destaca la incorporación de nuevas características X73PHD como la Métrica Permanente (*Permanent Metric*, PM) y el tipo Enumeración (*Enumeration*) al evolucionar la plataforma hacia adquisición y transmisión en tiempo real (aunque solamente disponible en las configuraciones extendidas de algunos dispositivos).

El modelado de la pila de protocolos X73PHD ha sido modificado para reducir la complejidad del programa e incorporar completamente la nueva máquina de estados FSM. Mientras que en las plataformas anteriores (basadas en PoC) estaban implementadas todas las capas del modelo OSI como funciones o clases con sus correspondientes variables locales y vinculaciones con otras clases, *plataforma2.0-release* (basada en PHD) procesa la cabecera de los mensajes y sigue un esquema de interpretación determinado según el tipo de servicio al que pertenece la cabecera. De esta manera, es posible acceder a la información contenida en la trama desde la aplicación central, como se detallará más adelante. En Fig. 3 puede verse un esquema comparativo de los dos modelos.

Los tipos de datos definidos en ASN.1 son codificados a MDER de manera más eficiente que en las versiones anteriores para reducir la complejidad de la plataforma. Se ha prescindido de librerías adicionales y los mensajes, objetos y datos son codificados y agregados a la trama *on-the-fly*. Esta estrategia descarga la complejidad de la etapa de ejecución del protocolo al tiempo que la desplaza hacia la etapa de desarrollo. Esto es debido a que el uso de punteros en C++ y la manipulación de *arrays* de bytes (tramas de transmisión) han de realizarse con extrema precaución para no provocar errores en tiempo de ejecución (desbordamiento de memoria, punteros a zonas restringidas, etc.)

El procesado de las tramas se hace de manera bastante similar a las plataformas anteriores aunque ahora se da un tratamiento más atomizado aprovechando las características de la codificación MDER. Este tipo de codificación explota el hecho de que los paquetes intercambiados contienen campos que se repiten constantemente (como cabeceras de tipo de protocolo, longitud o indicadores de segmento), mientras que los campos contenedores de datos de medidas se actualizan. Así surge la propuesta de uso de patrones de tramas, almacenados previamente en memoria que agilicen el procesado y eviten la necesidad de implementar demasiadas funciones, lo que permitirá la implementación del diseño propuesto en dispositivos basados en micro-controlador [20].

Como resultado de todas estas premisas, se muestra en Fig. 4 el esquema completo de la solución que implementa el módulo de comunicaciones X73PHD. Todas las características del protocolo, el modelo de capas, la gestión de tramas, la definición de tipos de datos, y el diseño de la máquina de estados, se encuentran encapsulados en varias clases (codificadas en C++ y usando librerías ASN.1), siendo `PHDappAgent` y `PHDappManager` las más representativas ya que implementan el nivel de aplicación del protocolo. Además, es importante remarcar que se cumple una de las restricciones de diseño orientado a entornos personales en el uso de memoria al haber reducido de unos 8450KB con X73PoC a 1880KB con X73PHD, además de rebajar el número de librerías y clases en un factor de casi 20, sin contar librerías de transmisión. Esto permite asegurar que el diseño basado en estándares no añade peso a la implementación final, lo que garantiza su implantación en dispositivos *wearables*.

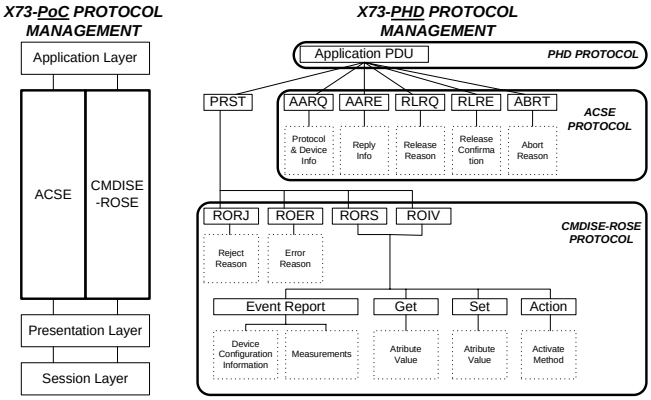


Fig. 3. Comparativa de modelos X73PoC y X73PHD

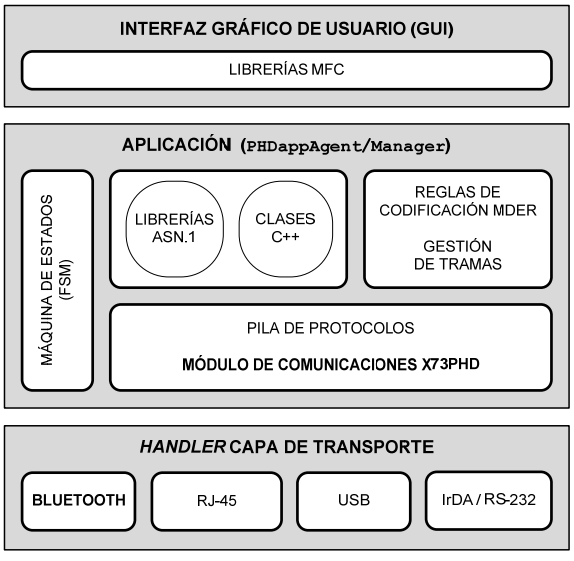


Fig. 4. Diseño basado en X73PHD. Módulos software necesarios

B. Requisitos conforme al estándar EN13606

EN13606 [7] se ha desarrollado para la representación de cualquier información incluida en la HCE, así como para su comunicación entre diferentes sistemas de información sanitaria, consiguiendo la interoperabilidad semántica de la información transmitida. El objetivo principal de este estándar es normalizar el intercambio de HCEs (completas o sólo ciertas partes, denominadas extractos) de forma que éstas sean interoperables. Por tanto, no se define la manera en que la información relativa a un paciente debe ser almacenada para ser consultada en un centro u hospital, sino sólo el modo en la que la información debe ser intercambiada.

Para ello, EN13606 se basa en un modelo dual: un modelo de referencia (que da soporte a la información) y un modelo de arquetipos (que define el conocimiento). Así si el conocimiento varía, solo variará el arquetipo bajo el que la información se transmite: por ejemplo, la medida de la presión arterial de un paciente se recogería en el modelo de referencia, mientras que el conocimiento asociado a que la presión arterial consta de sistólica y diastólica se recogería en el modelo de arquetipos. Con todo ello, el estándar está compuesto por 5 partes: 1-Modelo de Referencia, 2-Especificación de intercambio de arquetipos, 3-Arquetipos de referencia y listas de términos, 4-Características de seguridad, y 5-Modelos de Intercambio.

Aunque EN13606 no estandariza cómo han de ser almacenados los datos, para poder transmitir un extracto en relación a lo que la norma establece, sí se debe dar soporte a varios tipos de información atendiendo a cada caso específico (ver Fig. 5). A continuación se describen brevemente estos bloques lógicos en los que se estructura el sistema:

- *Extract* (extracto): hace referencia a la HCE de un paciente.
- *Folders* (carpetas): organización a alto nivel de la HCE (por ej.: estudios, episodios, etc.).
- *Compositions* (composiciones): cualquier interacción médico-paciente o documento médico (informes, resultados de pruebas, etc.).
- *Sections* (secciones): encabezamientos que reflejan el proceso de razonamiento o consulta (no tienen significado semántico pero sirven de ayuda para navegar a través del documento, por ej.: síntomas).
- *Entries* (entradas): afirmaciones clínicas sobre observaciones, instrucciones, etc.
- *Clusters* (agrupación): Estructuras de datos multipartes anidadas (tablas, series temporales, etc.).
- *Elements* (elementos): Nodos finales con datos simples (medida concreta, motivo de la visita, etc.).

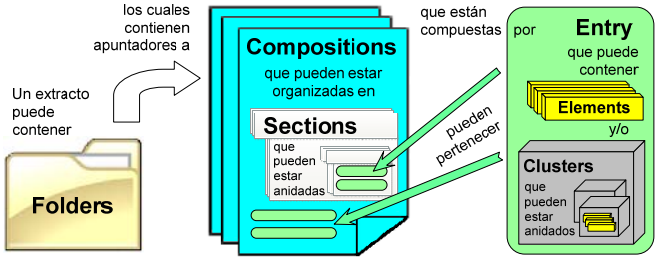


Fig. 5. Estructura del extracto de HCE según EN13606 (*extraído de* [7])

La situación actual del estándar es todavía inconclusa: de hecho, la Parte 5 no ha sido votada a inicios de 2009 y el modelo de referencia ha experimentado pequeñas variaciones con respecto a distintos versiones del pre-estandar ENV13606 (aprobado en 2004) flexibilizando la cantidad de conceptos que es necesario transmitir. Las principales diferencias son:

- La sensibilidad no es un parámetro obligatorio y ha pasado a ser un valor entero. La aparente paradoja que se crea, dado que la sensibilidad es una de las entradas dobles donde queda reflejado qué profesionales tienen acceso a qué información, queda resuelta por medio de la suposición de un valor de sensibilidad por defecto.
- El atributo que se usa para poder asociar distintas composiciones (*contribution_id*) pasa de la clase AUDIT_INFO a COMPOSITION, asociando este tipo de relación más como información clínica y no de contexto de toma de datos.
- La clase CLINICAL_SESSION desaparece siendo absorbidos sus atributos por las clases RECORD_COMPONENT y FUNTIONAL_ROLE; así COMPOSITION absorbe los atributos opcionales *sesión_time* (intervalo de tiempo que dura la interacción médico paciente) y *territory* (país en que se crea el extracto), y FUNTIONAL_ROLE absorbe *healthcare_facility* (clínica en la que se tomaron los datos) y *service_setting* (contexto de toma de medidas: domiciliario, hospital, etc.).
- Desaparece el atributo que representa al profesional sanitario que legalmente es responsable del paciente en el momento de almacenar los datos (*hca_legally_responsible_for_care*).
- Se sustituye el atributo optativo *composer*, por una asociación obligatoria a *committal* de la clase AUDIT INFO. De esta forma se obtienen mayor información de contexto al introducir quién y cuándo lo envía y desde qué sistema. Además esta nueva versión independiza quién lo manda y quién lo crea, que no tiene porque ser la misma entidad.
- Como en casos anteriores, la autoría de una composición viene determinada por *composer* pero, en esta versión, es un atributo de asociación de la clase FUNCTIONAL_ROLE, de la cual solo se obliga a transmitir el atributo *performer*.
- Por ultimo, de la clase LINK se ha eliminado el atributo *version_specific*, que indicaba si el objetivo era RECORD_COMPONENT o una versión. Dado que todas las versiones de RECORD_COMPONENT han de tener un identificador único, es lógico pasar ese identificador sin importar si corresponde a un registro sin versionar o versionado.

Tras el estudio de los campos que son obligatorios transmitir con la HCE y, debido a la herencia entre las consecuentes clases, se detallan en Tabla I el conjunto de campos mínimo necesario para representar cada uno de los elementos del modelo de referencia (indicando entre corchetes su tipo de datos) y su significado en la norma EN13606. Estos campos son los obligatorios en la transmisión y, si se recurre de forma más detallada a la norma, pueden utilizarse medios que permitan especificar información adicional (como que el extracto ha sido generado de manera automática entre dos máquinas, si la persona que autorizó la creación de ese extracto fue un determinado médico, o que se adjunta una prueba de lo que se visualizaba por pantalla a la hora de realizar la prueba como es obligatorio en muchos países).

TABLA I
CONJUNTO DE CAMPOS NECESARIOS EN EL MODELO DE REFERENCIA EN13606

EHR_ EXTRACT	Al enviar extracto de HCE se genera esta cabecera, tras la cual se transmiten las composiciones
ehr_id [*Instance Identifier*]	Identificador único del extracto (en un sistema de referencia y para el paciente en concreto)
ehr_system [*Instance Identifier*]	Identificador del sistema donde fue creado el extracto
rm_id [*String*]	Indicador de la versión del modelo de referencia (la norma establece el valor de "EN13606")
subject_of_care [*Instance Identifier*]	Identificador de paciente
time_created [*Time Point*]	Hora y fecha en que los datos de paciente fueron consultados o exportados para crear el extracto
EXTRACT_ CRITERIA	Parámetros opcionales (no obligatorios) a especificar en una petición/envío de extracto de HCE solicitado
all_versions [*Boolean*]	Indicador si se transmiten todas las versiones
archetype_ids [*Set Instance Id*]	Conjunto de arquetipos que fueron solicitados en la petición del extracto
max_sensitivity [*Integer*]	Sensibilidad máxima que se usó para generar el extracto
multimed_included [*Boolean*]	Indicador de si la información multimedia viaja con el extracto
other_constraints [*String*]	Otras restricciones del extracto
time_period [*Interval TimePoint*]	Periodo de tiempo para el que se define el extracto
RECORD_ COMPONENT	Clase abstracta que, por herencia, introduce estos atributos como obligatorios al resto de clases
name [*Text*]	Nombre del registro
rc_id [*Instance Identifier*]	Identificador único del registro en todo el sistema sanitario
synthesised [*Boolean*]	TRUE si RECORD_COMPONENT ha sido creado para cumplir con el estándar
COMPOSITION	Atributos heredados (no obligatorios) de RECORD_ COMPONENT más un atributo obligatorio *commital* de AUDIT_INFO, que contiene los siguientes atributos:
committer [*Instance Identifier*]	Quién genera RECORD_COMPONENT
ehr_system [*Instance Identifier*]	Desde qué sistema se genera RECORD_COMPONENT
time_committed [*Time Point*]	Hora y fecha en que se genera RECORD_COMPONENT
ENTRY	Atributos heredados de RECORD_COMPONENT
uncertain_expressed [*Boolean*]	Si los datos que contienen algún nivel de certeza
ITEM	Clase abstracta, por lo que es un caso particular, ya que no hay instanciación posible (aunque de ella derivan CLUSTER o ELEMENT). Introduce campos opcionales:
<*obs_time*>	Especificar el instante en que fue tomada la medida (una composición no tiene por qué crearse en el momento en que se recojan los datos)
<*emphasis*>	Indicar algún tipo de relevancia
CLUSTER	Atributos heredados de RECORD_COMPONENT, más:
structure_type [*Code SimpValue*]	Qué tipo de estructura es (los códigos se especifican en la parte 3 del estándar)
ELEMENT	*Clase hoja* ya que contiene los datos propiamente dichos
value [*Data Value*]	Un elemento tiene un campo simple DATA_VALUE que contiene el valor, a menos que esté indicado como ausente por medio de un atributo *null_flavour*

Por último, y dado que la forma de implementación no está definida específicamente por el estándar, para la implementación de EN13606 en una plataforma como la presentada en este artículo, sería necesario un repositorio de información donde almacenar composiciones a partir del cual se pueda construir el extracto de HCE conforme a la norma. En Fig. 6 se muestra en un ejemplo el envío de un extracto de HCE con un valor de la toma del peso de un paciente. Este ejemplo usa el modelo TS14796 que especifica el tipo de datos para CEN, dado que se está a la espera de la aparición de un nuevo conjunto unificado de datos para salud y que sea común a los estándares ISO, CEN y HL7.

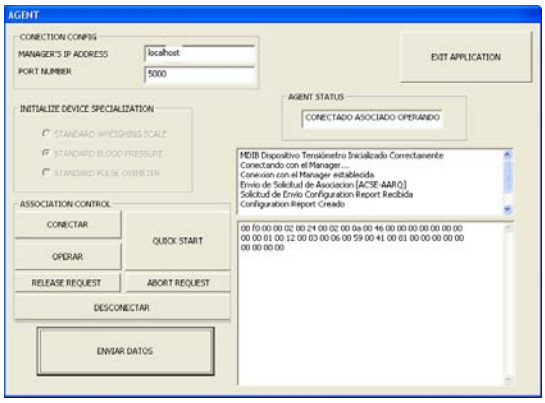
EHR_EXTRACT
ehr_system.extension = **HospitalServet**
ehr_system.assigningAuthorityName = **Salud**
ehr_system.valid_time = **1/1/1900 – 1/1/3000**
ehr_id.extension = **ExtractoHCE.120025022008**
ehr_id.assigningAuthorityName = **Salud**
ehr_id.valid_time = **1/1/1900 – 1/1/3000**
subject_of_care.extension = **441003686941**
subject_of_care.assigningAuthorityName = **Salud**
subject_of_care.valid_time = **1/1/1900 – 1/1/3000**
time_created.time = **15/02/2009 17:32**
rm_id = **EN13606-1.0**
COMPOSITION
rc_id.extension = **0003**
rc_id.assigningAuthorityName = **MiguelServet-Salud**
rc_id.valid_time = **1/1/1900 – 1/1/3000**
name = **Listado de datos de telemedicina**
sensitivity = **3**
committal.ehr_system.extension = **HospitalServet**
committal.ehr_system.assigningAuthorityName = **Salud**
committal.ehr_system.valid_time = **1/1/1900 – 1/1/3000**
committal.committer.extension = **Dr. Perez**
committal.committer.assigningAuthorityName = **Salud**
committal.committer.valid_time = **1/1/1900 – 1/1/3000**
committal.time_commited = **10/01/2009 17:32**
ENTRY
rc_id.extension = **0004**
rc_id.assigningAuthorityName = **MiguelServet-Salud**
rc_id.valid_time = **1/1/1900 – 1/1/3000**
archetype_id.extension = **CENArch.Entry.TMWeightMeasure.v1**
archetype_id.assigningAuthorityName = **MiguelServet**
archetype_id.valid_time = **1/1/1900 – 1/1/3000**
name = **Medida del peso**
meaning.codingScheme = **2.16.840.1.113883.6.96**
meaning.codingSchemeName = **SNOMED**
meaning.codingSchemeVersion = **7**
meaning.codeValue = **301333006**
meaning.displayName = **Medida del peso corporal**
synthesissed = **FALSE**
sensitivity = **3**
ELEMENT
rc_id.extension = **0005**
rc_id.assigningAuthorityName = **MiguelServet-Salud**
rc_id.valid_time = **1/1/1900 – 1/1/3000**
name = **Medida del peso**
meaning.codingScheme = **2.16.840.1.113883.6.96**
meaning.codingSchemeName = **SNOMED**
meaning.codingSchemeVersion = **7**
meaning.codeValue = **301333006**
meaning.displayName = **Medida del peso corporal**
sensitivity = **Clinical**
synthesized = **FALSE**
value.PQ.value = **77**
value.PQ.units = **kg**
value.PQ.property = **Weight**

Fig. 6. Esquema de extracto de HCE conforme a EN13606

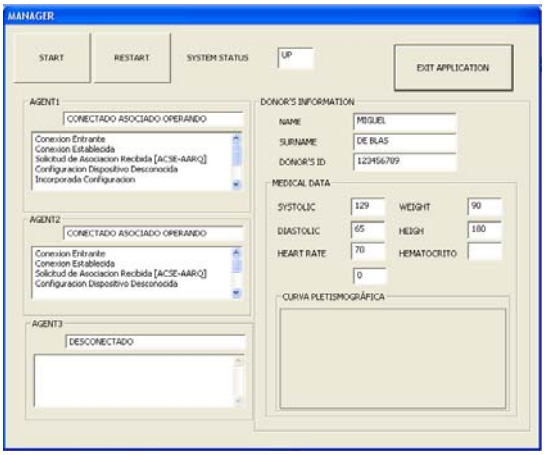
V. Resultados y puntos abiertos

Siguiendo las consideraciones de diseño e implementación, se presenta *plataforma2.0-release*: como solución multimedia basada en X73PHD/EN13606. En un entorno de investigación permite verificar la validez de los estándares y actualizar las nuevas especificaciones que las diversas versiones incorporen. En un entorno real (centros de salud, hospitales, etc.) permitiría integrar sistemas que, incluyendo interfaces estándares como USB o Bluetooth, no dan homogeneidad a las comunicaciones dado que no siguen un estándar [21]. Un ejemplo práctico de aplicación y su posible transferencia al sistema sanitario se muestra en Fig. 7 orientado a un banco de donantes de sangre y tejidos en el que se incluye un tensiómetro, un pulsioxímetro y una báscula. La aplicación *Agent*, como muestra Fig. 7(a), permite representar todas las especificaciones X73PHD para cada MD. La aplicación *Manager*, como muestra Fig. 7(b), permite monitorizar las conexiones de cada uno de los agentes y gestionar la transmisión de datos conforme a X73PHD.

Como retos inminentes, ya iniciados desde nuestro grupo de I+D, quedan por incluir dispositivos de adquisición de señales biomédicas en tiempo real (como ECG, cuyas especificaciones X73PHD acaban de aprobarse en 2009), la implantación en µcontroladores para dispositivos *wearables* en redes BAN, o la incorporación derequerimientos de seguridad computacional (confidencialidad, autenticación, integridad y, no repudio.



(a) Aplicación diseñada para el agente X73PHD (MD)



(b) Aplicación diseñada para el manager X73PHD (CE)

Fig. 7. Entorno gráfico del demostrador X73PHD de la plataforma

VI. Conclusión

La necesidad de integración en los diversos estándares médicos y su constante evolución en los últimos años, han derivado en la implementación de la plataforma presentada. El diseño propuesto garantiza la interoperabilidad de dispositivos de salud personal y su homogeneización con el HCE (conforme a los estándares de referencia X73PHD y EN13606, y un nuevo protocolo *end-to-end*, E2ESHP), y permite su inminente implantación en micro-controladores.

Agradecimientos

Los autores quieren agradecer las contribuciones a este trabajo de PHDWG y especialmente a Mr. Melvin Reynolds, *convenor* del CEN/TC251 WGIV; así como a Miguel Galarraga (investigador y profesor de UPNA) y Adolfo Muñoz (investigador del Instituto de Salud Carlos III, secretario de AENOR/CTN139, y representante español de CEN/ TC251) por sus excelentes contribuciones a esta investigación durante los últimos años. Este trabajo ha recibido el apoyo de la Comisión Interministerial de Ciencia y Tecnología (CICYT) y de los Fondos Europeos de Desarrollo Regional (FEDER) TIN2008-00933/TSI y TSI2005-07068-C02-01, del Ministerio de Industria, Turismo y Comercio TSI-020302-2008-35/Plan Avanza I+D, una beca FPI a M. Martínez-Espronceda (ref. 1342/2006 - UPNA), y una beca de investigación a J.D. Trigo (ref. IT7/08 - DGA/CONAID/CAI).

Referencias

[1] Pedersen S, Hasselbring W. Interoperability for information systems among the health service providers based on medical standards. *Inform Forsch Entwickl* 18(3-4):174-188, 2004.

[2] Comiteé European Normalisation/TechnComm251 (CEN/TC251). *www.centc251.org* (En España, AENOR/CTN139. *www.aenor.es/ desarrollo/inicio/home/home.asp*). [04/09].

[3] DICOM. *Digital Imaging and Communications in Medicine*. http://medical.nema.org/. Último acceso: 04/09.

[4] Fischer R. How to implement SCP-ECG? Parts I-II. *Hannover*, 2003.

[5] HL7. Devices Special Interest Group. *www.hl7.org/Special/committees/ healthcaredevices/index.cfm.* [04/09].

[6] Galarraga M. *et al*. Standards for medical device communication: X73 PoC-MDC. *Stud Health Technol Inform*. vol.121, pp.242-56, 2006.

[7] EN13606 CEN/TC251. Electronic Healthcare Record Communication. Standard Parts 1-5. *www.medicaltech.org.* [04/09].

[8] Integrating the Healthcare Enterprise (IHE). *www.ihe.net/*.[04/09].

[9] Continua Health Alliance. *www.continuaalliance.org/home/.* [04/09].

[10] ISO/IEEE11073 Point-of-Care MD Communication standard (X73-PoC). Health informatics. [Part 1. MD Data Language (MDDL)] [Part 2. MD Application Profiles (MDAP)] [Part 3. Transport and Physical Layers]. *www.ieee1073.org.* [04/09].

[11] ISO/IEEE11073 - Personal Health Devices standard (X73-PHD). Health informatics. [P11073-00103. Technical report - Overview] [P11073-104xx.Device specializations][P11073-20601.Application profile-Optimized exchange protocol]. *http://standards.ieee.org/.* [04/09].

[12] I. Martínez *et al*. Implementación integrada de plataforma telemática basada en estándares para monitorización de pacientes. *Jornadas de Ingeniería Telemática*, pp. 505-512, 2007.

[13] I. Martínez *et al*. Optimización de una plataforma telemática para monitorización de pacientes orientada a u-Salud y basad en estándares y Plug-and-Play. *Jornadas de Ingeniería Telemática*, pp. 505-512, 2008.

[14] Yao J, Warren S. Applying ISO/IEEE11073 standards to wearable home health monitoring systems. *J Clin Monit Comput* 19(6):427-436, 2005.

[15] Warren S *et al*. Lessons learned from applying interoperability and information exchange standards to a wearable point-of-care system. *Conf Distr Diagn Home Healthcare*. doi: DDHH.2006.1624807, 2006.

[16] Clarke M *et al*. Developing a standard for personal health devices based on 11073. *Conf Proc IEEE Eng Med Biol Soc*, pp.6175-6177, 2007.

[17] Martínez I. *et al*. Standard-based Patient Monitoring Platform for Ubiquitous Environments, *Int Conf IEEE EMBS*, pp. 1813-1816, 2008

[18] Martinez I. *et al*. Implementation of an end-to-end standard-based patient monitoring solution. *IET Commun* 2(2):181-191, 2008.

[19] Muñoz A. *et al*. Proof-of-concept Design and Development of an EN13606-based EHR service. *J Am Med Inform Assoc* 14:118-129, 2007.

[20] Martinez-Espronceda *et al*. Implementing X73: Proposal of two different strategic approaches. *IEEE Eng Med Biol Soc*, pp.1805-1808, 2008.

[21] Martínez I. *et al*. Recent Innovative Advances in Telemedicine: Standard-based Designs for Personal Health. *Special Issue Advanced Simulation and Innovative Design in Bioengineering - Int J Biomed Eng Techn* (Y. González and M. Cerrolaza Eds.), 2009.

Servicio de Selección de Noticias basado en Mashup de Contenidos con CMIS

José M. Jiménez, Guillermo Hernández
División I+D+i
Informática Gesfor (Grupo Gesfor)
Avda Manoteras, 32
28050 Madrid
jmjimenezt,ghernandezc@grupogesfor.com

Carlos Á. Iglesias, David Jiménez
División I+D+i
Germinus XXI (Grupo Gesfor)
Avda Manoteras, 32
28050 Madrid
cif, djimenezc@germinus.com

***Resumen*—El artículo presenta la propuesta y resultados del proyecto Contenidos a la Carta, que propone el uso de tecnología de mashups para la selección de contenidos (noticias), combinado con el uso de la especificación CMIS (*Content Management Interoperability Service*) para la integración de repositorios heterogéneos de contenidos. El artículo presenta la arquitectura propuesta, que define operadores de mashup específicos para contenidos, para lo que se ha extendido la herramienta de creación de mashups MyCocktail.**

***Palabras Clave*—CMIS, noticias, mashups, contenidos, REST**

I. Introducción

La web 2.0 con fenómenos como YouTube ha abierto sin duda una era de los contenidos, tanto en su creación como en su consumo. Han caído las barreras financieras, tecnológicas y culturales que limitan la creación de conocimiento. Sin embargo la "nueva" economía sigue obedeciendo a las viejas reglas y lo que se hace abundante se devalúa. ¿Dónde está ahora el valor? Posiblemente en la presencia en Internet, y en la capacidad de filtrado, la agregación y la remezcla de contenidos y servicios, así como en la conexión intelectual y emocional con los usuarios.

En un mundo donde más gente aporta datos y produce información y conocimiento, una gran parte de estos contenidos presentan escaso interés y/o calidad. La necesidad de localizar, combinar y posicionar los contenidos no está restringido a usuarios finales, sino que es también una necesidad empresarial para los proveedores de contenidos, que necesitan combinar sus contenidos propios y externos para poder recuperar la inversión realizada en su creación y añadirles valor.

El proyecto Contenidos a la Carta investiga y experimenta en técnicas y herramientas que faciliten la composición de ofertas personalizadas de contenidos, en este caso noticias. También investiga en técnicas y herramientas de posicionamiento de contenidos en español, así como técnicas de rastreo y protección de los contenidos en español en la red.

El proyecto se centra en el ámbito de las noticias de prensa y de la problemática de un proveedor de contenidos como la Agencia EFE, primera agencia de noticias en español y cuarta agencia mundial de noticias.

Para la composición de contenidos, el proyecto **Contenidos a la Carta** [2] investiga la aplicación del reciente estándar CMIS (*Content Management Interoperability Services*) [7] para proporcionar un servicio de interoperabilidad funcional entre los diferentes repositorios de contenidos de Agencia EFE. Mediante la aplicación de técnicas semánticas, permitirá ofrecer interoperabilidad semántica entre sus metadatos. El proyecto también investiga en la aplicación de técnicas de mashups de contenidos que permitan combinar y adaptar los contenidos para innovar en el proceso de composición de nuevos contenidos, mediante una interfaz gráfica de usuario.

Contenidos a la Carta pretende también investigar en el rastreo de contenidos en Internet para detectar copias. Este tema es altamente relevante para garantizar los derechos de los proveedores de contenidos. Aunque en el mercado existen productos comerciales, como Attributor, estos productos están en inglés y las adaptaciones al castellano son muy pobres.

El proyecto experimenta sobre nuevos métodos para incrementar la eficacia y la eficiencia de las empresas dedicadas a la creación, transformación y distribución de contenidos, aumentando su competitividad y aumentando la presencia de contenidos digitales de calidad en Español en Internet, al ofrecer nuevos canales de distribución de noticias innovadores y flexibles, adaptables a los contínuos cambios que aparecen en el mundo de la gestión de contenidos.

El resto del artículo se estructura como sigue. La sección II describe el estándar CMIS. A continuación, la sección III revisa el estado del arte en tecnologías de mashups y, en concreto, de la herramienta de creación de mashups MyCocktail. Posteriormente, en la sección IV se ilustra la solución propuesta mediante una descripción de la arquitectura y un caso de uso en la sección V. Por último, se recogen las conclusiones y trabajos futuros en la sección VI.

II. El estándar CMIS

El estándar CMIS (*Content Management Interoperability Services*, Servicios de Interoperabilidad de Gestión de Contenidos) [7] ha sido impulsado por IBM, EMC y Microsoft y respaldado por Opentext, Oracle, Alfresco y SAP, y viene a resolver uno de los mayores problemas que las empresas han acusado con respecto a la gestión de su información empresarial, que es la integración de repositorios de contenidos.

A través de un juego común de servicios, CMIS permite interactuar con los diversos repositorios de gestión de contenidos, sin importar quién es el fabricante del repositorio o cómo éste está implementado.

El objetivo de este estándar es permitir que las aplicaciones puedan trabajar con cualquier tipo de repositorio de

contenidos de manera uniforme y busca asegurar la interoperabilidad de las aplicaciones que usan múltiples repositorios de contenidos.

Para resolver tales problemas, CMIS define un modelo de dominio para interactuar con repositorios ECM (*Enterprise Content Management*) haciendo uso de Servicios Web. Provee un gestor de contenidos para modelos de datos de dominios específicos, un conjunto de servicios genéricos que actúan en ese modelo de datos y varios protocolos para acceder a esos servicios, incluyendo SOAP (*Simple Object Access Protocol*) y REST/Atom (*Representational State Transfer*) [6].

Dentro del modelo de dominio, CMIS define un modelo de datos, donde se especifican los elementos necesarios para trabajar con un gestor de contenidos, i.e. el repositorio y los objetos básicos que componen un repositorio: documentos, directorios, relaciones o la política administrativa. Además de definir con el modelo de datos los elementos del repositorio con los que operará CMIS, también es necesario definir en el modelo de dominio un conjunto de servicios genéricos que actuarán en ese modelo de datos. Estos servicios ofrecen las operaciones típicas de gestión de contenidos en un repositorio, tales como creación, búsqueda, edición o borrado de contenidos, conexión y desconexión a un repositorio, etc [7].

Dado que el objetivo del estándar es facilitar el intercambio de información y documentos entre entornos y repositorios documentales diferentes, eliminando los problemas de migración entre una plataforma y otra, y facilitando que coexistan sistemas de diferentes fabricantes (permitiendo la federación, por ejemplo), se hace necesario definir, además del modelo de dominio, una serie de API's de comunicación, orientadas a la definición de servicios Web, y un protocolo de publicación Rest/Atom que puede ser usado por aplicaciones para trabajar con uno o más gestores de repositorios de contenidos u otros sistemas. Los protocolos de comunicación empleados en el estándar son REST/Atom y SOAP Web services.

Para que sea posible ese intercambio de información entre repositorios documentales, es necesario que esos repositorios implementen el estándar CMIS. Alfresco, empresa participada por SAP, actualmente implementa esta especificación en su última herramienta lanzada al mercado.

Alfresco es un sistema de administración de contenidos de código abierto, que proporciona gestión de documentos, herramientas de colaboración, gestión de contenidos Web, además de otras muchas funcionalidades. La arquitectura de Alfresco está basada en tecnologías de código libre tales como Spring [23], Hibernate [20], Lucene [19], modernos estándares como JSR-168, JSR-170 [13], servicios Web, Java Server Faces [21] y contribuciones de la comunidad de software libre. Alfresco incluye un repositorio de contenidos, un framework de portal web para administrar y usar contenido estándar en portales, un sistema de administración de contenido, capacidad de virtualizar aplicaciones web y sitios estáticos vía Apache Tomcat, búsquedas vía el motor Lucene y flujo de trabajo en jBPM. Alfresco está desarrollado en Java [3].

Alfresco Labs3 [3], la última versión de código abierto de Alfresco lanzada al mercado, incorpora una serie de servicios Web basados en CMIS que cubren las funcionalidades especificadas en dicho estándar.

CMIS debería hacer que los ECM pudiesen relacionarse mediante la tecnología de mashups, construyendo aplicaciones más ricas y rápidas. En definitiva, se espera que CMIS se convierta en una revolución en la gestión de contenidos equiparable a la que supuso SQL en el mundo de las bases de datos [14].

III. Tecnología de Mashups

Los mashups son composiciones de contenidos y servicios obtenidos de diferentes fuentes que se presentan de manera homogénea ofreciendo un valor adicional respecto a los datos presentados por separado. En la web existe un número creciente de mashups que se ha visto acentuado por la aparición de APIs que facilitan la tarea de la reutilización del código [24].

Este aumento de los mashups junto con la tendencia cada vez mayor de la creación de contenidos por parte de los usuarios en Internet motiva la aparición de herramientas para la fácil creación y edición de mashups.

Según Gartner [10], los mashups han pasado del puesto 6 en 2008 al puesto 5 en 2009 en el ranking de tecnologías emergentes, destacando su penetración en las Empresas. Gartner predice que el 80% de las nuevas aplicaciones estarán basadas en mashups. Actualmente, hemos identificado las siguientes tendencias en tecnologías de mashups. Para cada una de ellas están surgiendo herramientas de creación y edición de mashups.

- *Mashups de Datos y Servicios* – Coleccionan datos y servicios de diferentes orígenes y los mezclan en una una interfaz gráfica común. Algunas herramientas para la creación de este tipo de mashups son: YahooPipes [17], Popfly [9], MyCocktail [22].
- *Mashups de Procesos* – Permiten combinar diversos procesos como alertas o envío de correos. En la actualidad existen pocas herramientas para la creación de este tipo de mashups y, a excepción de OPUCE [15], la mayoría son propietarias: Serena Software [25], K2-BlackPearl [16].
- *Mashups de Interfaz de Servicios* – Permiten la creación de gadgets de una forma sencilla, evitando la programación y utilizando interfaces gráficas usables por usuarios no muy experimentados en estos entornos. Morfeo Fast [8].
- *Mashups de Navegador* – Son aplicaciones en forma de plugin de navegadores que permiten extraer información de páginas web para ser combinada en forma de mashups. Permite consumir la información de las páginas de forma personalizada, enriqueciendo la experiencia del usuario. Debido a la novedad de este tipo de mashups las herramientas existentes son poco intuitivas y presentan muchas limitaciones. Intel Mash Maker [12], Ubiquity [26], y Piggy Bank [5] son algunos ejemplos.

A. Herramienta de Mashups MyCocktail

MyCocktail [22], Romulus Mashup Builder, es una aplicación web que proporciona al usuario una interfaz gráfica para construir mashups fácilmente, permitiendo al usuario

desarrollar mashups de manera más rápida, incrementando así la productividad.

Esta herramienta permite al usuario combinar información proveniente de diferentes servicios, que puede ser modificada con operadores y más tarde presentada con una gran variedad de renderizadores. Todo este proceso se lleva a cabo mediante una interfaz gráfica de usuario de fácil manejo, que permite combinar componentes arrastrando y soltando. El tiempo que lleva desarrollar un mashup se reduce así considerablemente y se mejora la productividad.

MyCocktail permite a los diseñadores y programadores usar servicios sin preocuparse de detalles de bajo nivel. Los usuarios solo tienen que manejar una serie de herramientas que se proporcionan y MyCocktail hará las peticiones a los diferentes servicios.

MyCocktail está basado en Afrous [18] y proporciona tres tipos distintos de componentes, que combinados dan lugar al mashup:

- *Servicios*. Pueden invocarse varios servicios REST por defecto, como del.icio.us, Yahoo Web Search, Google AJAX Search, Flickr, Twitter, Amazon, etc.
- *Operadores*. La información obtenida se puede procesar por medio de operadores. Por ejemplo, es posible ordenar, filtrar o agrupar infromación según parámetros.
- *Renderizadores*. La información se puede presentar de diversas formas: HTML, diagramas estadísticos (gráfico de tarta o de barras), Google Maps.

Los pasos que suelen seguirse para contruir un mashup son:

1) El usuario obtiene información de uno o varios servicios.
2) Los datos obtenidos pueden filtrarse y procesarse usando los operadores proporcionados por la herramienta para extraer información útil para el mashup.
3) La información resultante puede mostrarse en HTML, en diagramas estadísticos o en mapas usando los renders.
4) Se exporta el mashup en uno de los diversos formatos que ofrece MyCocktail: JavaScript, HTML, Google Gadget o Netvibes Gadget.

La figura 1 ilustra el aspecto general de la herramienta de mashups. En ella se pueden observar dos partes bien diferenciadas. En la izquierda tendríamos un ventana donde se listan los servicios disponibles, los operadores y los renders. En la ventana de Servicios, el usuario obtendría contenidos a través de llamadas a los servicios Web de los Sistemas de Gestión de Contenidos y la de los principales portales Web2.0 como Flickr, Google Maps, del.icio.us, Twitter o buscadores como Google y Yahoo.

Los operadores permitirían realizar operaciones con los resultados obtenidos de los servicios. Con estos operadores el usuario podrá realizar mashups de contenidos. Cabe destacar que se incluirán operadores específicos para noticias. Por último los renders nos permitirán mostrar y publicar el resultado en varios formatos, entre ellos HTML, Javascript o como un Gadget. La otra parte que forma la herramienta de mashups será la ventana central donde se desplegaran los operadores, renders o las llamadas a los servicios.

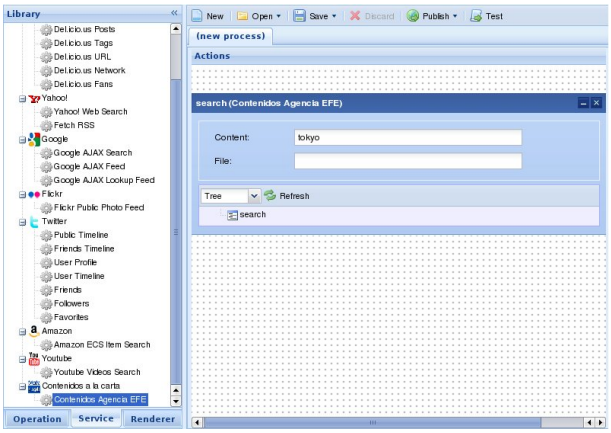


Fig. 1. Herramienta de Construcción de Mashups MyCocktail

IV. Arquitectura de Contenidos a la Carta

Esta sección introduce brevemente el proyecto en el que se enmarca la investigacion, Contenidos a la Carta, así como la arquitectura propuesta para la realización de mashups de contenidos.

La meta de este proyecto es mejorar la capacidad de ofrecer noticias personalizadas a determinados tipos de usuarios y automatizar determinados procesos relacionados con la publicación de noticias, como pueden ser la traducción, la difusión y la protección de los contenidos relativos a la noticia. Esto supondrá un avance significativo en la industria de los medios de comunicación, ya que permitirá a éstas disponer de unos contenidos más ricos y de mayor calidad, a la par que aumenta las posibilidades de difusión de sus noticias a lo largo de la Internet. Además, la plataforma de Contenidos a la Carta pretende conseguir que el uso de este tipo de sistemas se extienda en todas las empresas de este sector, ya que se requieren pocos conocimientos técnicos para la composición de noticias personalizadas y de gran calidad.

Contenidos a la Carta es, por tanto, un proyecto orientado a la investigación de métodos de selección, personalización y difusión de noticias mediante la reutilización de diversos contenidos multimedia alojados en sistemas de gestión de contenidos heterogéneos a través del estándar CMIS. Esto será posible gracias a la plataforma de Contenidos a la carta, que proporcionará herramientas de tipo mashup, adaptables a las necesidades y requerimientos de cada usuario, para integrar y componer noticias a partir de diferentes orígenes de información. Para poder utilizar contenidos de sistemas de gestión de contenidos diferentes de una manera flexible, en el marco de proyecto se tiene previsto desarrollar una capa de homogeneización de servicios de gestión de contenidos. Esta capa de homogeneización será posible a través del estándar CMIS.

La arquitectura del proyecto Contenidos a la Carta está formada, por tanto, por una herramienta de Mashups y por uno o varios sistemas de gestión de contenidos. De esta forma, el usuario tendrá ante si una plataforma donde podrá recuperar contenidos de fuentes heterogéneas a través de servicios basados en CMIS, podrá trabajar con ellos formando noticias personalizadas a través de los operadores específicos para noticias que se han diseñado en la herramienta de mashups y los publicará en diferentes formatos de representación, ya

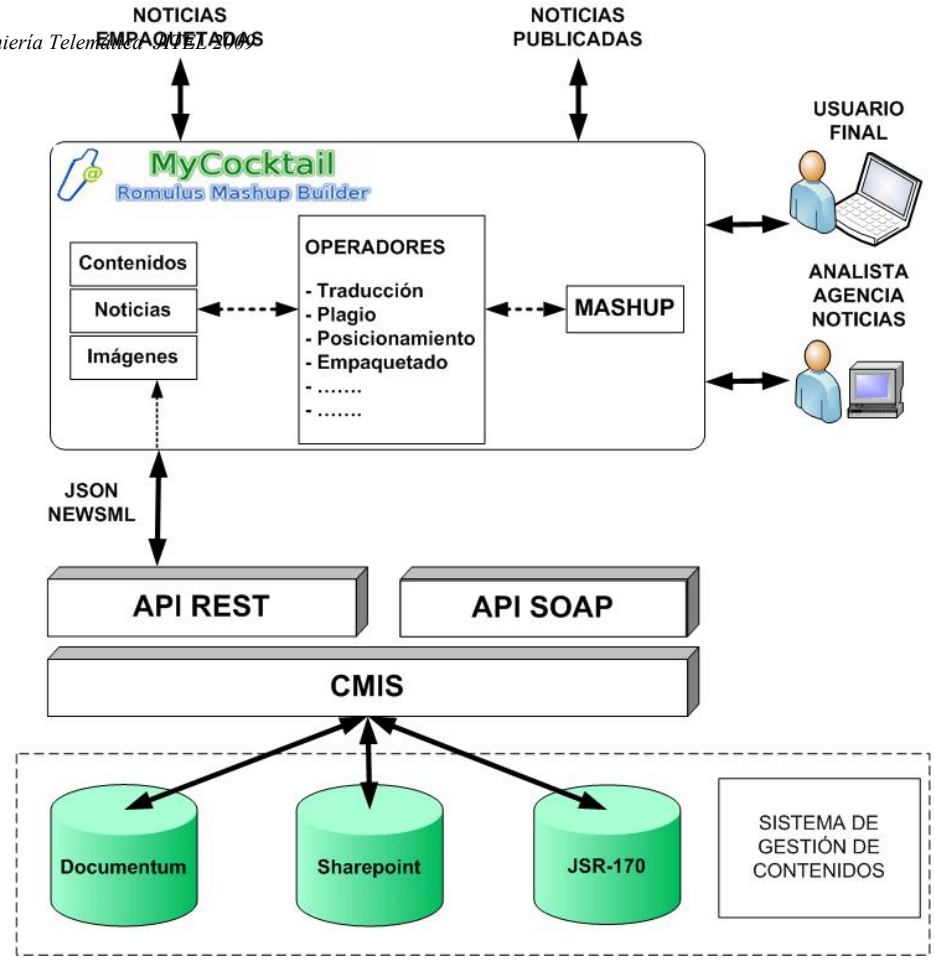


Fig. 2. Arquitectura de Contenidos a la Carta

sea HTML, XML, etc. En la figura 2 podemos observar el esquema de la arquitectura general de Contenidos A la Carta.

Uno de los dos elementos principales de la arquitectura de Contenidos a la Carta son los *Sistemas de Gestión de Contenidos*. La idea principal del proyecto es comunicar estos sistemas con la herramienta de mashups, de tal manera que podamos operar con los contenidos almacenados en los diferentes repositorios y crear nuevos contenidos mezclando algunos de ellos.

Dado que el proyecto está enfocado a la experimentación en la mejora del proceso de oferta personalizada de noticias, es importante tener en cuenta que los contenidos que se manejarán en la herramienta de mashups y se almacenarán en los repositorios serán noticias, por lo que se han querido adaptar los modelos de tipos de contenidos de los Sistemas de Gestión de Contenidos a algún estándar de noticias, que favorezca el intercambio de noticias como NewsML.

NewsML [11] es un estándar desarrollado por el International Press Telecommunications Council (IPTC [1]) que define un formato común para el intercambio de noticias, sin tener en cuenta el tipo de contenido multimedia en el que se presenta la noticia. Este intercambio de noticias es un método que permite no solo transmitir los contenidos de noticias, sino además describir el contenido de manera abstracta, mediante metadatos, con información relativa acerca de cómo manejar las noticias de una manera apropiada, así como su empaquetado e, incluso, el tipo de transmisión que se llevará a cabo.

NewsML es un lenguaje de contenedores de noticias digitales. Es decir, con NewsML no pueden crearse noticias en un formato concreto: es preciso disponer de ellas previamente en otros formatos. Su utilidad es transportar paquetes de contenidos periodísticos, sea cual sea su formato y su forma de difusión: texto, imagen y sonido. NewsML, permite contener más de un paquete informativo (o, simplificando, más de una noticia) en un solo documento. La estructura básica de un documento NewsML se muestra en la figura 3.

```
<NewsML>
    <NewsItem>
        <NewsComponent >
            <ContentItem>
                (Aquí puede el contenido de una pieza
                informativa, en cualquier formato, o una
                referencia a ese contenido, que esté físicamente
                en otro fichero.)
            </ContentItem>
        </NewsComponent >
    </NewsItem >
</NewsML>
```

Fig. 3. Formato NewsML

Básicamente:

- Cada pieza está incluida en un elemento ContentItem.
- El conjunto de varios ContentItem (un texto, una foto, un vídeo y un gráfico) pueden formar parte de un NewsComponent (una noticia).
- Un NewsItem puede contener varias noticias, varios NewsComponent.

Un documento NewsML puede llevar además toda una serie de metadatos que, en resumen, pueden ser de tres tipos:

- Datos relativos a la transmisión del documento NewsML en conjunto: quién lo envía, a quién, cual es su prioridad, la fecha de caducidad, etc.
- Datos sobre el documento en cuestión, o sobre partes determinadas: por ejemplo, en cada una de las piezas que se incluye, pueden añadirse elementos que describan el tema, o los protagonistas, o las relaciones de esa pieza con otras en el mismo documento, etc.
- Datos sobre cómo se normalizan los datos incluidos en otros elementos de NewsML. Por ejemplo, si en un NewsItem se incluye una noticia sobre fútbol, se puede incluir un elemento, o Topic, que describa temáticamente el deporte sobre el que trata la noticia. Ese elemento puede incluir un código; será entonces, un elemento añadido, o Catalog, el que indique cual es la clasificación o vocabulario del que forma parte el código, y dónde se encuentra la clasificación completa, en Internet.

La mayor parte de los metadatos que forman parte de NewsML pueden situarse en múltiples lugares del documento, aplicados a una sola pieza o a varias. Como puede deducirse, NewsML tiene dos características básicas:

- Una estructura modular, que además permite que las piezas estén situadas físicamente dentro del documento NewsML o fuera de él, unidas mediante referencias a objetos externos.
- La posibilidad de realizar una descripción estructural, aunque sea solo para distinguir cada una de las piezas (pero no sus partes), y una descripción semántica, también de tipo general.

La herramientas de mashup permite a los usuarios la creación de aplicaciones que manejen las fuentes de datos de manera sencilla y flexible, dando la posibilidad de aplicar un amplio conjunto de operadores que transformarán y combinarán los contenidos. Las entradas de estos operadores, que a partir de este momento denominaremos *"pipe"*, pueden ser la salida producida al procesar una o varias entradas por parte de otro operador o un servicio de datos proporcionado por la capa de interoperabilidad CMIS.

Los servicios que ofrece la capa de interoperabilidad pueden ser servicios Web XML o servicios REST. A las salidas de datos, además de generar *pipes*, también se les puede aplicar un proceso denominado renderización de contenidos, que no es más que aplicar un proceso que produce una salida visualizable del contenido por algún tipo de cliente estándar (Navegador Web, Navegador Móvil). Mediante las operaciones de renderización se permitirá seleccionar el tipo de dispositivo sobre el que se visualizará el contenido.

La comunicación entre los Sistemas de Gestión de Contenidos y la herramienta de mashups se realizará mediante peticiones por parte de la herramienta de mashups a los servicios Web de los Sistemas de Gestión de Contenidos para recuperar contenidos. La herramienta de mashups se encargará de buscar y recuperar contenidos en forma de noticia, procesarlos y permitir al usuario que realice un nuevo contenido a partir de la mezcla de varios de ellos.

La herramienta de mashups tiene definida como entrada ficheros JSON, que posteriormente procesa y devuelve el resultado en diversos formatos, por lo que es necesario que las respuestas de los servicios Web de los repositorios ECM sean en formato JSON. En caso de que alguno de dichos repositorios no soporte trabajar con formatos JSON, será necesario incluir un conversor de este formato.

Este es el caso de los Sistemas de Gestión de Contenidos Alfresco Labs3 utilizados en Contenidos a la Carta. En este caso, se deberá incluir un conversor, como Apache Abdera [4], para la traducción de atom/xml (el formato utilizado por Alfresco Labs3) a JSON. La elección de Alfresco Labs3 como Sistema de Gestión de Contenidos para Contenidos a la Carta es debido a que es uno de los primeros gestores de contenidos de código abierto que implementa el estándar CMIS.

Dentro del proyecto, hemos identificado los siguientes operadores para la realización de mashups de contenidos:

- **Consulta**. Los servicios de consulta de noticias diseñados constan de un operador de búsqueda textual de contenidos, un operador de consulta avanzada basada en metadatos y un operador de búsqueda multilingüe, donde el usuario tendrá la posibilidad de buscar noticias en diversos idiomas.
- **Operaciones de posicionamiento**. La herramienta de mashups dispondrá de un operador de extracción de metadatos de la noticia y sugerencia de términos para su posicionamiento en buscadores (SEO, *Search Engine Optimization*, logrando que las noticias tengan un mayor alcance y difusión. Este operador permite automatizar la tarea de catalogación y extracción de palabras clave, lo que en el contexto de las noticias resulta fundamental ya que permite evitar demoras en la publicación de las mismas.
- **Rastreo y protección de contenidos**. El uso de este operador servirá para detectar copias de las noticias, garantizando así los derechos del autor de las noticias. Este operador utilizará técnicas de detección de plagio comparando el mayor o menor grado de coincidencia entre fragmentos de las diferentes obras según diferentes propiedades, tales como la frecuencia de palabras, el uso de un tipo u otro de palabras o aparición de erratas.
- **Traducción de noticias**. Este operador facilitará al usuario la traducción de noticias, tanto del texto que la forma como los metadatos de la noticia, lo que permitirá realizar búsquedas multilingües.
- **Empaquetado**. Una vez generado un "pipe", este operador empaquetará el conjunto de contenidos de noticias, ya sea artículos de texto, imágenes, vídeo o cualquier tipo de contenidos multimedia, en un package para su difusión y exportación a NewsML. La exportación a este formato de intercambio de noticias, junto a los operadores de posicionamiento y traduccion, hacen que la difusión de las noticias sea mucho mayor que la obtenida utilizando los métodos de publicación tradicionales.

Por último, cabe destacar que la plataforma Contenidos a la carta integra un módulo que permite generar aplicaciones adaptables bajo la tecnología de Mashup, también se permite a los usuarios seleccionar el formato de visualización que se aplicará a los contenidos mediante filtros de renderización, esto permitirá que los usuarios puedan visualizar los contenidos en distintos dispositivos. Esto quiere decir que la in-

formación del contenido se puede separar de la representación del contenido.

V. Caso de uso

En esta sección se desea mostrar al lector un ejemplo de aplicación y uso de Contenidos a la Carta, para que vea las mejoras que puede ofrecer esta plataforma al proceso de creación de noticias personalizadas por parte de las agencias de noticias.

Anteriormente, cuando hablábamos de la arquitectura de la plataforma señalabamos que utilizaríamos el estándar NewsML-G2 para modelar los diferentes tipos de contenidos que soportará el repositorio. Esto es así porque las agencias de noticias hacen uso de este formato estándar de intercambio de noticias para la creación y definición de contenidos. Contenidos a la Carta aprovechará este formato de intercambio de noticias, además de la tecnología CMIS, para automatizar, en la medida de lo posible, todo el proceso de creación de una noticia y centralizar dicho proceso a través de una única plataforma.

En este ejemplo nos situaremos en el papel de un periodísta que desea realizar un artículo sobre destinos turísticos en Italia. Para crear dicho artículo, nuestro periodísta utilizará una serie de contenidos que tiene a su disposición, como son una serie de reportajes de texto sobre distintas ciudades y algunas fotografías tomadas por los fotografos de la agencia. La secuencia de pasos que el periodísta deberá seguir estará compuesta por la búsqueda de los artículos de texto e imagenes, la reunión de dichos contenidos en una misma noticia y por último, subirlo al repositorio de gestión de contenidos, donde la noticia, aprovechando las funcionalidades de los repositorios ECM, podría ser revisada por un supervisor y, posteriormente, apobar su publicación. Mediante la plataforma de Contenidos A la Carta todas estas operaciones pueden ser realizadas desde un mismo interfaz, mediante el cual se podrá operar con los contenidos almacenados en los repositorios de contenidos, creando nuevas noticias a partir de ellos y dárlas de alta en el sistema.

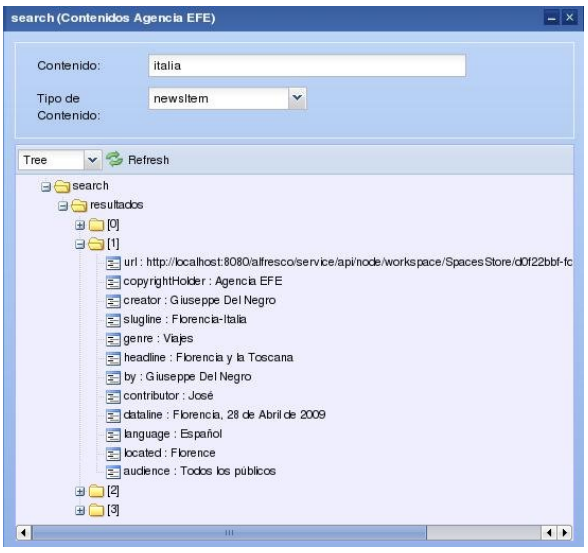


Fig. 4. Búsqueda de Noticias

Como hemos comentado, el primer paso del periodista será la búsqueda, en el repositorio de la agencia, de contenidos o noticias relacionadas con destinos turísticos en Italia. Esto queda ilustrado en la figura 4.

Mediante este servicio de Búsqueda de Noticias, la plataforma realizará una llamada al repositorio ECM, apoyandose en el estándar CMIS, buscando coincidencias entre la palabras clave que el usuario ha introducido y los metadatos de las noticias. El repositorio de contenidos devolverá una lista, en formato JSON, de las noticias que contiene algún metadato relacionado con las palabras clave que el usuario ha introducido.

Una vez que el periodista ha seleccionado algunos artículos con los que le gustaría construir el reportaje, podrá utilizar el servicio de empaquetado de noticias para ello. Mediante este servicio se podrá construir, a partir de varios contenidos, una noticia completa publicable. En la figura 5 podemos ver que el usuario podrá indicar los metadatos del nuevo reportaje. Una vez que el servicio se lleve a cabo, el sistema automáticamente dará de alta el reportaje en el repositorio donde, como hemos dicho anteriormente, podrá entrar en una cadena de revisión y publicación por parte de revisores y analistas.

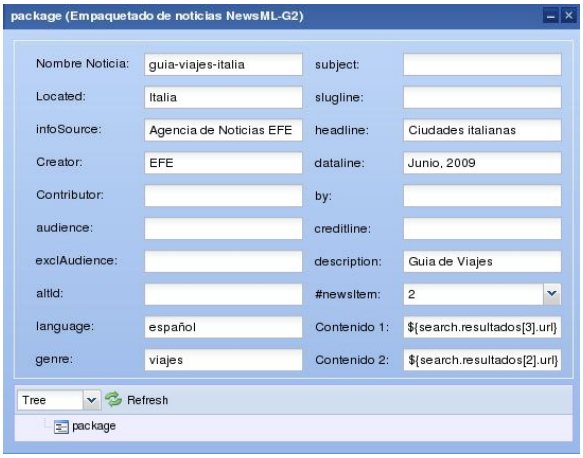


Fig. 5. Empaquetado de Noticias

Las noticias creadas mediante la plataforma de Contenidos A la Carta pueden crear rápidamente noticias personalizadas ricas en contenido mediante una sola plataforma. Otra de las ventajas es que además de aprovechar los recursos propios de la agencia, desde la misma plataforma se podrán acceder a contenidos de sitios 2.0 típicos, tales como vídeos de Youtube, fotos de Flickr, comentarios de Twitter, etc.

Como ejemplo de esta capacidad de aprovechar todas las posibilidades que ofrece Internet, en la figura 6 se muestra cómo el usuario puede utilizar sus contenidos y, por ejemplo, la API de Google Maps para realizar, mediante uno de los servicios definidos en la plataforma, un mapa interactivo con las noticias (en este caso reportajes) distribuidas por su lugar de origen.

VI. Conclusiones y Trabajos Futuros

En este trabajo de investigación se ha presentado el proyecto Contenidos a la Carta, basado en tecnologías de mashups e interoperabilidad de contenidos para la composición de ofertas personalizadas de contenidos.

El trabajo ha definido la arquitectura del proyecto, formada, principalmente, por una herramienta de Mashups y por uno

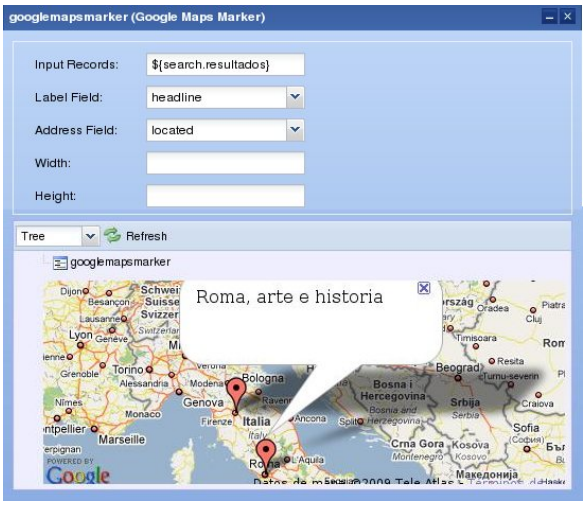


Fig. 6. Mashup de contenidos del repositorio y Google Maps

o varios sistemas de gestión de contenidos. La tecnología de mashups se ha postulado como una de las tecnologías emergentes en la gestión y personalización de contenidos. Los operadores de contenidos para realizar mashups presentados están ofreciendo resultados interesantes, y grandes posibilidades si los combinamos con otros operadores existentes. Por ejemplo, podemos filtrar noticias y mostrar estadísticas gráficas de autores o mostrar las noticias en un mapa según el lugar de la noticia.

Actualmente Contenidos a la Carta se encuentra en fase de desarrollo y tiene también la intención de lograr potenciar el posicionamiento de las noticias en buscadores, mediante técnicas SEO (*Search Engine Optimization*) [27].

AGRADECIMIENTOS

Este proyecto ha sido financiado por el Ministerio de Industria, Turismo y Comercio, dentro de la convocatoria 2/2008 del subprograma Avanza I+D, como proyecto de Desarrollo Experimental (TSI-020501).

REFERENCIAS

[1] International press telecommunications council, consorcio que agrupa a las más importantes agencias de noticias y empresas de comunicación, disponible en http://www.iptc.org/.
[2] Web de contenidos a la carta. disponible en http://alacarta.germinus.com, 2008.
[3] Alfresco. Alfresco labs3, un sistema gestor de contenidos de código abierto, disponible en http://www.alfresco.com, 2008.
[4] Apache. Web del proyecto apache abdera, una implementación abierta de atom, disponible en http://abdera.apache.org/, 2009.
[5] P. Bank. Piggy bank, 2009.
[6] R. Cover. Content management interoperability services (cmis), información disponible en http://xml.coverpages.org/cmis.html. Technical report, OASIS, 2008.
[7] EMC Corporation, IBM Corporation, and Microsoft Corporation. *Content Management Interoperability Services, borrador del estándar disponible en http://www.alfresco.com/about/cmis/cmis-draft-v0.5.zip*, 8 2008.
[8] M. FAST. Morfeo fast, disponible en http://fast.morfeo-project.eu/, 2009.
[9] M. P. Fly. Microsoft pop fly, disponible en http://www.popfly.com, 2009.
[10] Gartner. Information technology research and advisory company, disponible en http://www.gartner.com, 2009.
[11] K. Holland. IPTC Standards: EventsML-G2 version 1.1, NewsML-G2 version 2.2, SportsML-G2 version 2.0. Guide for Implementers. Technical report, IPTC Standards. International Press Telecommunications Council, 2009.
[12] I. M. Maker. Intel mash maker, disponible en http://mashmaker.intel.com, 2009.
[13] S. Microsystems. Jsr 170: Content repository for java technology api, disponible en http://jcp.org/en/jsr/detail?id=170, 2009.
[14] J. Newton. Hacia la estandarización ecm con cmis, artículo disponible en http://www.techweek.es/gestion-documental/opinion/1003918003401/estandarizacion-ecm-cmis.1.html, Noviembre 2008.
[15] P. Opuce. Proyecto opuce, disponible en http://www.opuce.tid.es/, 2009.
[16] K. B. Pearl. K2 black pearl, disponible en http://www.k2.com/en/displaycontent.aspx?id=903, 2009.
[17] Y. Pipes. Yahoo pipes, disponible en http://pipes.yahoo.com, 2009.
[18] A. Project. Afrous project web site, disponible en http://afrous.com/, 2009.
[19] A. L. Project. Apache lucene project web site, disponible en http://lucene.apache.org/java/docs/, 2009.
[20] H. Project. Hibernate project web site, disponible en http://www.hibernate.org/, 2009.
[21] J. Project. Javaserverfaces project web site, disponible en http://java.sun.com/javaee/javaserverfaces/, 2009.
[22] R. Project. Mycocktail web site, disponible en http://www.ict-romulus.eu/web/mycocktail, 2009.
[23] S. Project. Spring project project web site, disponible en http://www.springsource.org/, 2009.
[24] SalesForce. Mashups: The what and why, 2007.
[25] S. Software. Serena software, disponible en http://www.serena.com/, 2009.
[26] Ubiquity. Ubiquity, disponible en http://labs.mozilla.com/projects/ubiquity/, 2009.
[27] Wikipedia. Posicionamiento en buscadores, 2009.

Metadata Negotiation for the Optimization of Energy consumption in Wireless Sensor Networks

Sury Bravo L.[1], Marta Zuazua R.[2], José Fernán Martínez O.[3], Ana Belén García H. [4], Iván Corredor[5]
Departamento de Ingeniería y Arquitecturas Telemáticas DIATEL. Universidad Politécnica de Madrid.
E.U.I.T. Telecomunicación. Ctra. Valencia, Km 7 28031 – Madrid
Teléfono: +34913365526
E-mail: {sbravo[1], mzuzua[2], jfmartin[3], abgarcia[4], icorredor[5]}@diatel.upm.es

***Abstract*- Wireless Sensor Networks are composed by hundreds or even thousands of sensorial nodes that are scattered in an area, with capacities of observation, data processing, independent decision making and development of actions in answer to the measurements of the sensors and the information shared among them. They have restricted capacities of processing being communication the main consumer of energy. There are many researches that suggest different approaches regarding the Energy consumption management. In this work we present a proposal that combines the advantages of the network protocol MMSPEED with metadata negotiation, thus allowing decreasing unnecessary data transmission to the destination by means of deciding if it is necessary to make a retransmission or not based on the registered data and its calculations of reliable connections. This proposal is innovating as it skips a number of significant retransmissions and therefore improves nodes energy efficiency.**

Keywords.- Wireless Sensor Networks, Metadata Negotiation in WSN, Energy Consumption in WSN.

I. INTRODUCTION

In [1] Wireless Sensor Networks (WSNs) are defined as "large populations of wirelessly connected nodes, capable of computation, communication, and sensing. Sensor nodes cooperate in order to merge individual sensor readings into a high-level sensing result, such as integrating a time series of position measurements into a velocity estimate." There are other elements such as Gateway and the Base Station that are part of WSNs.

The basic elements of WSNs are sensor nodes that are responsible for capturing information from the environment in which they are deployed. The nodes are composed of volatile memory (RAM), read only memory (ROM), CPU, short-range radio module and power source. Additionally the most advanced sensing features that we can mention are: temperature, pressure, sound, humidity, lighting, etc.

One of the key constraints of the nodes is the energy source consumption since they have to combine autonomy with the capacity to process. The node energy source restricts the network functionality and reduces life-time. In [2] the parts that require more energy consumption are identified as: control unit, the radio module and transducers translating physical magnitudes to electric signals. On the other hand, there are many researches that suggest different approaches regarding the energy consumption management. We can highlight: model-aided metadata management [3] and Data-Centric routing [4] .

One more thing to take into account is the traffic of the generated data, which has an important redundancy, as sensors which are close to each other are going to generate the same type data. The routing protocols must take advantage of this situation in order to improve the consumption of energy. Among these protocols we can mention: SPIN (Sensor Protocols for Information via Negotiation) [5], SPMS (Shorted Path Minded SPIN) [6] and MMSPEED (Multipath Multi-SPEED Protocol) [7].

II. RELATED WORKS ABOUT OPTIMIZATION IN MANAGING THE ENERGY CONSUMPTION IN WSN

In [3] five approaches about metadata (data description) management are evaluated from two points of view related to the cost of energy and the reliability. The energy cost focuses on the diffusion of queries to the sink nodes and the transmission of the metadata from the nodes to the sink nodes. The reliability focuses on evaluating the errors between the metadata given by the sink nodes and the metadata from the sensor nodes. The approaches consuming more energy were two. The first approach reported metadata to the sink nodes periodically and the second, approach reported metadata only when they have significant variability. The approach consuming less energy gathered metadata only if a query command is received.

In [4] the strategies of dissemination centered on data and communication protocols, regarding how these minimize the energy consumption, are analyzed. As a result a mechanism to find the critical value of "interested nodes" to make the decision of how to transmit the data is proposed. This concept is important as it can result, both when designing the network and during its operation, in an increment of the efficiency in the energy consumption. This study is grounded on the use of the family of protocols SPIN and Shorted Path Minded SPIN (SPMS), which use routing techniques based on metadata.

SPIN is a family of protocols based on the negotiation of metadata, before transmitting it. In the negotiation mechanism, the nodes must describe the sent data using metadata, so the size of data frame is reduced; and hence the energy consumption is decreased.

The negotiation consists of which the nodes must describe the sent data using metadata that are shorter than the data being described. SPIN uses three types of messages: ADV, REQ, and DATA. For spreading information, the node sends

and ADV message with the corresponding metadata. If a neighbouring node is interested in the information it responds with a REQ message and then the node that generates the information responds with a DATA message that is the one containing the data. By these means SPIN guarantees that useful data will be sent and received, by request and without redundancy since in this negotiation it uses smaller metadata that the original data.

SPMS is an extension of SPIN that uses low-power, multi-hop communication for the REQ and DATA messages. SPMS geographically divides the entire sensor network in multiple zones based on the sensor nodes' transmission range. First stage is to transmit broadcast with ADV packet to the entire neighbouring zone. If a node is interested in the data, it will send a REQ packet to the one that originated the data. REQ packet is sent through the shortest path, using the lowest energy transmission in every hop. If the node is not just one hop from the source, it will have to send its REQ message through multiple hops. If the destination realizes it has to do multi-hop communication for its REQ packet, it waits for a pre-determined fixed period of time before sending the request packet [5] [6] .

In [7] , a Multi-Path and Multi-SPEED Routing Protocol (MMSPEED) for probabilistic QoS guarantee in WSN is proposed. The provided QoS is performed in two quality domains, namely, timeliness and reliability. Multipath routing solutions provide the reliability domain, by sending the same packet through different paths depending on the required delivery probability; assuring the arrival of data to the sink. The timeliness domain provides multiple network-wide speed levels; that can be selected by the type of traffic in a dynamic way, based on their end-to-end deadlines.

MMSPEED protocol needs adaptations on the MAC layer in order to handle prioritized access to the shared medium, multicast and the support of measurements of average delay and loss rate. It supports the IEEE 802.11e MAC protocol minor changes enough to handle the different priority level.

Shorter inter-frame spacing leads to a higher probability of high priority packets to get access to the shared medium.

As described in [8] [9] , the MMSPEED protocol has deficiencies in the aggregation network, the energy-delay trade-off and the facility for parameter interchange with MAC layer. On the other hand, Z-MAC protocol is proposed as an alternative to IEEE 802.11e. Although this protocol needs several additional features to be completely compatible with MMSPEED, it is an excellent base since it implements a priorities mechanism that is very appropriate for this study case. Additionally the use of metadata combined with this protocol in order to eliminate the data redundancy, generated by nodes MMSPEED, is proposed as a future focus research.

In [3] metadata in a WSN are defined as "the descriptive data used to describe the WSN system, including the environment, the nodes and their states, measurement data, and the WSN as a whole entity".

In [10] metadata is defined as information about information and their usefulness is described:

- Summarizing the meaning of the data.
- Allowing users to search for the data.
- Allowing users to determine if the data is what they want.
- Giving information that affects the use of data, and
- Indicating relationships with other resources.

As showed in the previously described literature, there is a relationship utility of metadata in negotiation, aggregation and data transmission in the WSNs. Therefore this relationship could be capitalized on the inclusion of metadata features in the protocol MMSPEED to improve its efficiency in the data transmission, avoiding redundancy and optimizing energy consumption. In this sense the next chapter deals with a proposal that addresses this problem.

III. Analysis of the Energy Consumption in Metadata Negotiation

In [5] the benefits of the protocol of data aggregation that we have previously mentioned are analyzed: SPIN (Sensorial for Protocols Information via Negotiation).

The SPIN family of protocols incorporates two key innovations to overcome the deficiencies of the classic flooding of the disseminating data such as implosion, overlap, resources blindness.

Following are the main functions the SPIN Protocol:

- Negotiation: before transmitting data the nodes negotiate with others to avoid the implosion and the overlapping.
- Only useful information is sent.
- Metadata must describe the data
- Adaptation of resources:
    - Each sensorial node has its manager of resources
    - The applications drill the manager before sending or processing resources.
    - The sensors can reduce certain activities when the energy is low.

Diverse statistical studies have demonstrated that protocol SPIN reduces to a 3.5% the energy consumption. This protocol has two ways of working: SPIN1, which would be the normal way and SPIN2, where depending on a predefined threshold, a node can change its way of performance and stop sending information.

The classes of messages that SPIN uses would be divided into three:

- ADV: the node announces to the network that it has certain data by means of the use of metadata.
- REQ: it is a message used to ask the node for sending the wanted data announced in the previous ADV.
- DATA: the data themselves.

On the other hand MMSPEED provides QoS differentiated in two dominions, timeliness and reliability. In the timeliness dominion a certain speed can be assigned to each packet of the different deliveries. In the timeliness domain a different sending speed can be assigned to each packet.

$$Speed_{i,j}^{k} = \frac{dist_{i,k} - dist_{j,k}}{delay_{i,j}} \quad (1)$$

Where $dist_{i,k} - dist_{j,k}$ is the distance from the source node *i* to the origin *k*, passing through the node *j* and $delay_{i,j}$ is the time that takes for a packet to go from the node *i* to the node *j* for its destination.

This protocol provides multiple layers of speed to use a virtual overlapping, consisting of virtual layers, as if it had two different networks that, in fact, is the same physical network, which implies that some nodes can belong to different levels of speed. In order to achieve that, each sensor node has a classifier which places the packets in the corresponding queue, with the required speed of transmission,

assigning the most appropriate speed for the packet, based on the distance towards its destination. This speed is given by:

$$\mathrm{Re}\, qSpeed(x) = \frac{dist_{s,d}(x)}{deadline(x)} \quad (2)$$

The classifier of the source node gathers the layer of higher speed choosing the node whose estimation speed is greater than *SetSpeed.*

$$SetSpeed_l = \min_{j=1}^{L}\{SetSpeed_j \,|\, SetSpeed_j \geq \mathrm{Re}\, qSeed(x)\} \quad (3)$$

When congestion exists, the packets with lower speed are discarded, guaranteeing that the packets with higher speeds survive.

The reliability dominion makes multipath forwarding based on the local estimation and the dynamic compensation. In order to control the number of routes by which the packet will be forwarded, MMSPEED offers a differentiation service: if the packet has a low level of required reach, it will be sent by a single route, and if the packet has a high level of required reach, it will be sent by multiple routes.

Each node locally determines the multiple links for forwarding the information based on the local estimation of the error and geographic hop distances to immediate neighbours.

$$RP_{i,j}^{d} = (1 - e_{i,j})(1 - e_{i,j})^{[dist_{j,d}/disti,j]} \quad (4)$$

Where $e_{i,j\ j}$ is the considered probability of loss of packets, including both intentional removing of packets, by control of congestion, and errors in the wireless channel.

By means of this estimation of reach probability of the packet via node *j*, the number of forwarding routes that satisfy the requirement with limited ranges to end of the packet can be determined by:

$$TRP = 1 - (1 - \mathrm{TRP})(1 - RP_{i,j}^{d}) \quad (5)$$

The expression obtains the set of required forwarding nodes, the packet is delivered to them using MAC multicast service.

The dynamic compensation is used to solve the erroneous calculations that will result from calculations locally carried out in each node which will go on forwarding the packet to the destination.

This protocol overloads the nodes with the decision making, such as the accomplishment of calculations of control and determination of neighbouring candidate nodes to pass the packets that offer to assure the model QoS.

The massive delivery of packets to support the reliability and the multi-speed layer mechanism (low or high reliability) can generate a significant data redundancy and other kind of overload in the node, using excessively its resources.

IV. Discussion and proposal of optimization of the energy management for the metadata negotiation

Since we have seen in previous sections, in the WSN, the motes have too much restricted energy resources and bandwidth of wireless connections to connect to them, being this one of the facts that causes more cost of energy.

A routing algorithm determines the path of a packet from a source to a destination. This path is due to think with base to objectives, like the maximum time of life of the network, the certainty that all the messages arrive at the sink node or minimum overload of the network. MMSPEED provides a high reliability in the arrival of the messages to the sink node, being this one of its kindnesses. Nevertheless, during its procedures to achieve reliability, it presents a considerable data redundancy.

Our research proposes a service for metadata negotiation that will use the services of MMSPEED along with the collected data to negotiate with the neighbouring nodes and to determine if a node that has registered an event has a route reliable enough so that it initiates the data transmission towards his destination. When two nodes in the network are about to deliver a message, they carry out two main processes: one to decide which routes will be use and the action or actions to be taken when a packet is received and forwarded. At this moment our metadata negotiation service (see Figure 1) would start to act as we explain next:

STEP 1:

The application layer captures the event registered by the sensor node, which is passed to the metadata negotiation service that will generate a Protocol Data Unit (PDU) with:

- The registered data
- The calculations of reliability that MMSPEED makes
- The number of required connections to obtain this reliability

STEP 2:

The PDU of the metadata negotiation service is encapsulated in the PDU of MMSPEED, being distinguished as a multicast PDU, and corresponding to the metadata negotiation service "MNS". This PDU will go as well encapsulated in the PDU of the MAC layer.

STEP 3:

The transmission of the packets of metadata negotiation begins. The neighbouring nodes will receive a PDU that contains the metadata, and then the negotiation will be initiated by MNS.

STEP 4:

The metadata negotiation service will first compare and check that everything in the node data agrees with the negotiated metadata, and if it does not, the negotiation will be discarded.

If the node data agree with the negotiated metadata, it continues comparing with the reliability, the number of required connections to reach this reliability and the distance to the sink node (calculated from the coordinates that each node has to its neighbours). Once the best option is chosen, that is the one with better reliability, requires less number of connections and has shorter distance to the sink than the other nodes involved in the negotiation; this node will be the one who initiates the transmission of the registered data.

It will probably be required to make cache of data while the negotiation takes place. This cache will be determined by the capacity of the processor and the speed of the physical channel.

This metadata negotiation service should be used for negotiating metadata transmission between neighbouring one hop nodes, since in these nodes shared redundant information is more probable.

Hence, the impact of the multicast is minimum, the packets will not travel beyond neighbouring nodes, since most likely they will have already registered the information, being therefore immediately processed. On the contrary if the packets have not registered the data, they will be discarded.

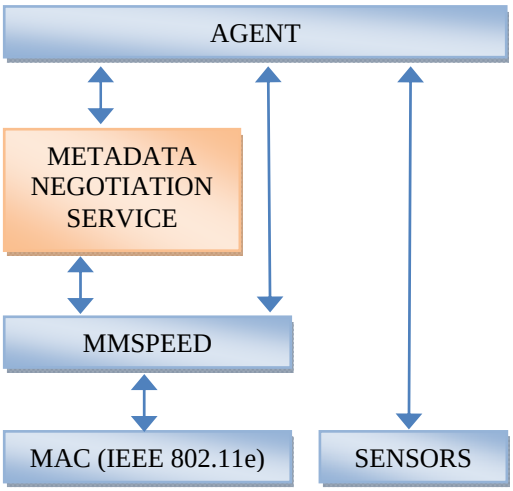


Fig.1. Proposed Metadata Negotiation Service

According on the architecture proposed in Figure 1 for MNS, the agent will obtain the sensor nodes data; it will calculate the required probability of reaching by means of MMSPEED services, based on the content of the data. Next the metadata negotiation service will determine, by means of the comparison of its metadata with the data of the node or involved neighbouring nodes in the negotiation, which node must initiate the transmission of the packet carrying the registered data. Since one of MMSPEED kindness is the delivery of its own packet by different routes, we can exploit that characteristic, by applying metadata negotiation only on reliable routes. Thus we can avoid transmitting the data if one of those nodes already has registered it. MMSPEED uses the IEEE 802.11e standard in the MAC layer, since it is appropriate for *ad hoc* sensor networks, and nodes can be developed and work in an *ad hoc* way without any special access points. As in IEEE 802.11e standard, the prioritization is achieved by differentiating inter frame spacing (IFCS) and Backoff Intervals for different classes.

V. Overhead Analysis

We propose a Wireless Sensor Network deployed in the environment to monitor temperature events and to inform if something abnormal happens to the temperature, that is to say, if it has exceeded a threshold. Each node must know its geographic position according to its neighbouring nodes, for which the sensor nodes usually use low cost mechanisms, with minimum energy consumption and that offer reliability when finding nodes within the network. Amount these mechanisms we found the Global Positioning System (GPS) [11] and location with base to directionality [12]. Other alternatives techniques to calculate the node position based on distances are: Iterative ML and Collaborative ML [13].

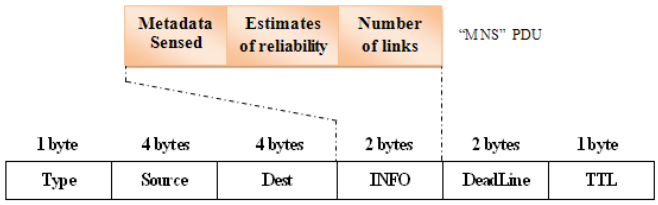


Fig.2. MMSPEED PDU encapsulating MNS PDU

Let us consider that we have two types of traffic: one from real-time events and another from monitoring. For instance, when an event is generated, it is detected by one node and its neighbours *j1* and *j2*. Each one of them makes the PDU in the metadata negotiation service layer with the metadata from the registered data, its reliability calculations and the number of necessary connections to reach this reliability. Then, the PDU is labelled as *Metadata Negotiation* and transmitted by Multicast mode (see figure 1) using the services of MMSPEED protocol which PDU is shown in figure 2. As well, the MMSPEED protocol uses the services of a MAC layer based on IEEE 802.11e protocol as it was mentioned. In our scenario, the nodes that are involved in event detection (*s*, *j1* and j2) initiate a multicast transmission with our own calculations. According to the data received from its neighbours, the *Metadata Negotiation* layer of each node starts the negotiation, with the following data:

Node	Reliability calculation	Number of connections	Distance to the sink
S	0,88	2	100 m
j1	0,90	1	80 m
j2	0,51	2	70 m

Table 1. Calculations which are used for negotiation previous to the data transmission

As we see in table 1, node *j1* is the one with better reliability and it requires a single connection to obtain it, even though it is a bit further from the sink node than node *j2*. It is important to say that larger amount of connections implies larger data transmission and this is one of the WSN characteristics that compromise energy resources; so node *j1* will be the one that must initiate the data transmission towards node *d* (sink).
With the metadata negotiation in the example of figure 3 two packets of the registered information that would arrive at the sink *j1* via *j3* towards the node was sent from the sink node *d*. The delivery of three metadata packets is necessary to negotiate which node makes the transmission of the data (see Table 2).

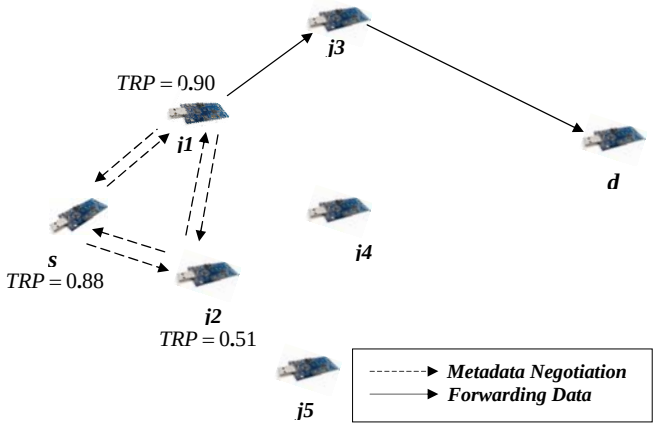


Fig.3. Forwarding with Metadata Negotiation Service

Packet from the node	Forwarding to d	Forwarding metadata	Number of forwarding packets
s		1	0
j1		1	0
j2		1	0
j1	j3, d	0	2
Total forwarding (data and metadata)		**3**	**2**

Table 2. Data which is negotiated in the transmission

On the other hand, without the metadata negotiation (see figure 4) fifteen packets were send to the sink *d*, since node s

and its neighbours j1 and j2 have registered the same information, and each one of them decides to transmit their data (see Table 3).

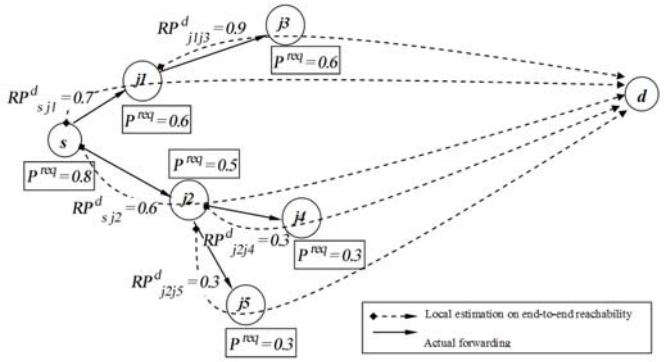


Fig.4. Forwarding with MMSPEED [7]

The metadata negotiation decreases data redundancy presented by protocol MMSPEED, by means of the use of a metadata packet that will only be interchanged between nodes which take part in the negotiation. The use of multicast packets guarantees that the metadata packets arrive only to neighbouring nodes, which are probably those having registered the same event depending on the proximity to their occurrence.

Packet from the node	Forwarding to d	Forwarding metadata
s	j1, d	3
	j2, j4, d	3
	j2, j5, d	3
j1	j3, d	2
j2	j4, d	2
	j5, d	2
Total forwarding (data)		**15**

Table 3. Data sends without metadata negotiation

VI. Conclusion and Future Work

Wireless sensor networks are an important type of resource-constrained distributed event-based system. As we have seen there are several networks protocols for this kind of sensors: some of them work with data aggregation and others with multi-hop sending. We identified and investigated some of the factors affecting performance, such as the communication and the routing decisions.

The protocol MMSPEED proposes a solution that tackles several QoS aspects, however the measure of energy consumption has not been taken into account, and this is a very important feature in the network life time (quite limited in WSN). With our proposal we improve the transmission power cost, through the metadata negotiation in the routing decisions.

Our service of metadata negotiation saves energy consumption in the node, with which the traffic in the network is decreased and bottlenecks in the transmissions are eliminated. Consequently the data transmission that has been registered by one or more near nodes is avoided, and data redundancy that arrives at the sink is decreased. The sensor nodes situated near the sink node are the most problematic ones because they absorb all traffic from the different parts of the network. Therefore, if we achieve to decrease transmissions in a global way in the network, those nodes could save energy significantly. Obviously, with our proposal the nodes nearest the sink node will be beneficed since they will decrease the transmission of packets originated by the data redundancy.

On the other hand, a general solution that decreases the energy consumption in a wide variety of scenes based on metadata negotiation does not yet exist. Therefore, we consider that our proposal means a great step in that sense. Since this is only a theoretical proposal, we are currently working in a case of study to demonstrate it- and to achieve a global solution.

Acknowledgements

This work has been partially funded by the Spanish Ministry of Industry, Tourism and Trade (“Ministerio de Industria, Turismo y Comercio”) in the framework of the AVANZA I+D programme, under the project named “Sistemas Inteligentes sobre Redes de Sensores Inalámbricos para el cuidado del Medio Ambiente” (ITEA2 – ESNA “European Sensor Network Architecture”), id. TSI-020400-2008-127.

References

[1] Elson, J. and Römer, K. 2003. Wireless sensor networks: a new regime for time synchronization. SIGCOMM Comput. Commun. Rev. 33, 1 (January 2003), pp. 149-154. DOI= http://doi.acm.org/10.1145/774763.774787

[2] Itziar Marín, Eduardo Arceredillo, Aitzol Zuloaga, and Jagoba Arias. Wireless Sensor Networks: A Survey on Ultra-Low Power-Aware Design. In Proceedings of world academy of science, engineering and technology volume 8, (Budapest, Hungary, October 2005), ISSN 1307-6884. http://www.waset.org/pwaset/v9/v9-1.pdf

[3] Chongqing Zhang, Haibing Guan, Minglu Li, Min-You Wu, Wenzhe Zhang and Feilong Tang. Model-Aided Metadata Management for Wireless Sensor Networks, Lecture notes in computer science. May 2006.

[4] Xuan Zhong, Ravish Khosla, Gunjan Khanna, Saurabh Bagchi and Edward J. Coyle. Data-Centric Routing in Sensor Networks: Single-hop Broadcast or Multi-hop Unicast?. Vehicular Technology Conference, 2007. VTC2007-Spring.IEEE 65th, (Dublin), April 2007.

[5] W. Heinzelman, J. Kulik, y H. Balakrishnan, “Adaptive Protocols for Information Dissemination in Wireless Sensor Networks”, Proc. 5th ACM/IEEE Mobicom Conference (MobiCom ’99), Seattle, WA, August, 1999. pp. 174-85

[6] Khanna, G. Bagchi, S. Yu-Sung Wu, “Fault tolerant energy aware data dissemination protocol in sensor networks”, Dependable Systems and Networks, 2004 International Conference on, pp. 795-804, 28 June-1 July 2004.

[7] E. Felemban, C.G. Lee and E. Ekici “MMSPEED: Multipath Muti-SPEED Protocol for QoS Guarantee of Reliability and Timeliness in Wireless Sensor Networks”, IEEE Transaction on mobile Computing, Vol. 5, No. 6, pp. 738-754, June 2006.

[8] Martínez, J., García, A., Corredor, I., López, L., Hernández, V., and Dasilva, A. 2007. QoS in wireless sensor networks: survey and approach. In Proceedings of the 2007 Euro American Conference on Telematics and information Systems (Faro, Portugal, May 14 - 17, 2007). EATIS '07. ACM, New York, NY, 1-8. DOI= http://doi.acm.org/10.1145/1352694.1352715.

[9] Martínez, J., García, A., Corredor, I., López, L., Hernández, V., and Dasilva, A. 2007. Modeling QoS for Wireless Sensor Networks. IFIP International Federation for Information Processing. pp. 143-154. December 2007.

[10] Steinacker, A.; Ghavam, A.; Steinmetz, R., "Metadata standards for Web-based resources", Multimedia, IEEE, vol.8, no.1, pp.70-76, January-March 2001.

[11] N. Bulusu, J. Heidemann, D. Estrin, “GPS-less low cost outdoor localization for very small devices”, Technical report 00-729, Computer Science department, University of Southern California, April 2000.

[12] Nasipuri, A. and Li, K. 2002. A directionality based location discovery scheme for wireless sensor networks. In Proceedings of the 1st ACM international Workshop on Wireless Sensor Networks and Applications (Atlanta, Georgia, USA, September 28 - 28, 2002). WSNA '02. ACM, New York, NY, 105-111. DOI= http://doi.acm.org/10.1145/570738.570754

[13] Savvides, A., Park, H., and Srivastava, M. B. 2002. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proceedings of the 1st ACM international Workshop on Wireless Sensor Networks and Applications* (Atlanta, Georgia, USA, September 28 - 28, 2002). ACM, New York, NY, pp. 112-121.

AZIMUTH ROUTING FOR LARGE-SCALE WIRELESS SENSOR NETWORKS

Paweł Kułakowski
Department of Telecommunications
AGH University of Science and Technology
kulakowski@kt.agh.edu.pl

Joan García-Haro
Department of Information Technologies and Communications
Polytechnic University of Cartagena
joang.haro@upct.es

***Abstract*- In this paper, geographic routing protocols for wireless sensor networks are considered. While a combination of the greedy and face routing is a solution with a good performance well documented in the open literature, it requires the procedure of network graph planarisation which makes the protocol vulnerable to network topology changes and consumes extra energy. We introduce a new protocol called *azimuth routing* that is able to operate without any pre-routing procedures. We validate it through computer simulations. The obtained results prove its good performance and low routing cost.**

***Index Terms*- Wireless sensor networks, geographic routing, MAC and routing protocols.**

I. INTRODUCTION

In wireless sensor networks, nodes usually need to be aware of their positions to properly report events or phenomena of their interest. Naturally, the information about node positions in a wireless network can be exploited to enhance the performance of routing protocols. This stands behind the idea of *geographical routing*: a popular approach to the routing issue in wireless sensor and ad hoc networks.

The most straightforward concept of the geographical routing is its *greedy* version: if a node would like to send a packet to a sink (destination) which is out of its range, the node chooses as a forwarder its neighbour node which is closest to the sink. This extremely simple concept is also very effective, but only in dense networks. When the node density is low, there is a large probability that a packet is stuck in a network local minimum, i.e. in a node that has no neighbours closer to the sink. There are many variants of the greedy routing, an overview can be found e.g. in [1, 2], but all of them are not suitable for low-density networks.

An algorithm that guarantees the packet delivery is *face routing*, first described in [3] and later developed in many papers, e.g. [4]. A network graph is divided into empty zones (called *faces*) surrounded by the nodes and network edges (connections between the nodes). A packet is routed around a zone to find a node which is the closest to a sink. Then, the packet skips to a next zone and the process continues. Face routing requires the network graph to be planar, i.e. it cannot contain any crossing links. To perform the graph planarisation, each node should contact all its neighbours and the proper cooperative decisions should be made which network connections must be switched off. When the network topology changes (e.g. due to a sensor energy depletion, nodes mobility or a wireless channel fading), the planarisation procedure should be repeated.

The motivation for our work reported in this paper was to develop a geographic routing protocol that could operate efficiently without any pre-routing procedures like planarisation. Such an approach has multiple advantages. First, the routing protocol is more robust to nodes mobility and topology changes. Second, the energy is not wasted for transmitting and receiving extra packets. Finally, in some cases, e.g. in military applications, we would like to avoid the network activity unless an event occurs that will force the sensor network to react.

In this paper, we assume that each sensor node knows its position (geographical coordinates) and the position of the nearest sink[1]. Sensors may be not aware of their neighbours, however we assume there is a MAC protocol that enables to choose the forwarding node according to the routing decision, what is additionally discussed in Section II.F.

We propose a new protocol called *azimuth routing*. Azimuth routing uses the greedy procedure as long as it is possible. If there is a zone in the network without sensors and a node processing a packet has not any neighbours closer to a sink, the packet is forwarded along the borderline of the zone. However, there is neither initial planarisation procedure, nor any *faces* defined. The packet is forwarded to a neighbour that is the best according to a group of criteria taking into account relative positions of the forwarding node, its neighbours and the sink.

The rest of the paper is organized as follows. In Section II, azimuth routing is deeply explained. The performance of the proposed protocol is validated through computer simulations in Section III. Section IV concludes the paper. In the Appendix, the mathematical calculations that need to be performed by sensors are shown.

II. ALGORITHM DESCRIPTION

The general idea of the azimuth routing can be summarised as follows: if we encounter the zone in the network that is not covered by sensors, let's try to skirt it in the most promising direction. However, due to the possible

[1] It could be accomplished e.g. if there exist super-nodes (sinks, anchors) that broadcast strong beacon signals which allow all sensor nodes to calculate their coordinates. Yet, these topics are out of the scope of this paper.

different network topology arrangements, numerous cases need to be considered.

Basically, the azimuth routing protocol works according to three states that can be described as: *pure greedy*, *greedy/azimuth* and *pure azimuth*. In each state, a node that wants to send a packet chooses the best forwarder from among its neighbour nodes, according to some criteria, explained in the subsections below. The choice of the forwarder is made with the aid of extra data transmitted in the packet header. This data is as follows:

a. the positions of two previous nodes, i.e. the nodes that hold the packet before the actual transmitting node,
b. the least distance to the sink that have ever been reached during the process of forwarding that packet – we call it *least distance value*,
c. a single bit that holds the information about forwarding direction (clockwise or counter-clockwise), so called *direction bit,*
d. a single bit that holds the information if the forwarding direction has been changed or not, so called *change bit*.

Apart from this data, each node knows its own position and the position of the nearest sink. We use the euclidean metric to measure all the distances. In the three following subsections, we explain the decision process in all three routing states.

A. *Pure greedy state*

In the *pure greedy* state, the routing protocol proceeds like in a standard greedy routing. A neighbour with the least distance to a sink is chosen as a forwarder. As long as it is feasible, pure greedy routing is continued. Note that in this state, any extra data transmitted in the packet header are not required to the choice of the next forwarding node. If a packet is transmitted in pure greedy state from its origin node, the least distance value is always the distance to the sink of the actual node, while the direction bit and change bit are not defined. However, it may happen that an area not covered by sensors exists in the middle of the network and there is no node closer to a sink than the actual node. Then, the distance to the sink of the last node is remembered as the least distance value and the routing algorithm switches to *greedy/azimuth* state.

B. *Greedy/azimuth state*

In the *greedy/azimuth* state, the transmitted packet is forwarded along the borderline of the area without sensor nodes. At the beginning, when the *pure greedy* procedure is not possible, a node with the least azimuth angle relative to the direction to a sink is chosen as a forwarder, as it is shown in Fig. 1. This first hop determines the *forwarding direction* of the packet (clockwise or counter-clockwise) around the nearest sink and this data is stored in the *direction bit*. We can say that the packet starts to circulate around the sink looking for the way to reach it. In each next hop, there are three possibilities. First, a neighbour node can be found that is closer to the sink than it is indicated by the *least distance value*. In this case, such a node is a new forwarder, the least

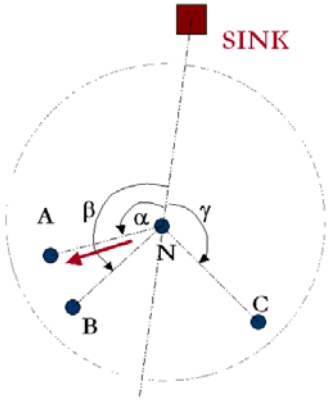


Fig. 1. The node N has just received a packet according to the *pure greedy* state, but it cannot find any greedy forwarder. Thus, the node N chooses a neighbour with least azimuth angle relative to the direction to a sink. As $\alpha < \gamma < \beta$, the node A is chosen as a forwarder. This determines the *forwarding direction*: in tis case, the packet will be circulating clockwise around the sink.

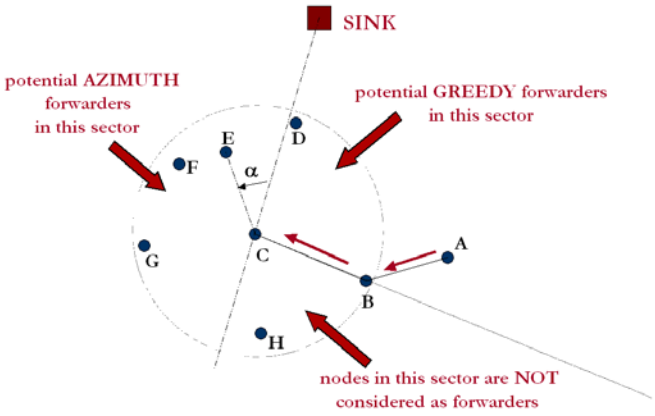


Fig. 2. The node C has just received a packet from the node B. It is looking for a forwarder according to *greedy/azimuth* state. In the presented situation, the node D will be chosen as a forwarder and the routing algorithm will switch to the *pure greedy* state. If the node D didn't exist, the node E would be chosen, because it would have the least azimuth angle to a sink (angle α). The nodes F and G have larger azimuth angles. The node H is not located in the proper sector and is not considered as a forwarder. The circle indicates the range of the node C.

distance value is updated and the protocol switches to the pure greedy state, again. Otherwise, the actual node investigates its neighbours looking for potential *greedy* forwarders, as shown in Fig. 2. Greedy forwarders are the nodes that are closer to the sink than the actual node, but the forwarding direction is changed by sending the packet there. If a greedy forwarder is found, the protocol also switches to the pure greedy state. Finally, if there is no greedy forwarders, the algorithm looks for a neighbour node which has the least azimuth angle to the sink and is located according to the forwarding direction (Fig. 2). Then, the greedy/azimuth state is continued until the sink is reached (Fig. 3).

In the worst case, it may occur that the actual node hasn't any neighbours that meet the abovementioned criteria (Fig. 4). Then, the forwarding direction is changed to the opposite one, this fact is noted in the *change bit* and the procedure is continued. If the lack of forwarders occurs again, that is after certain amount of hops there are again no possible forwarders, the algorithm switches to the last possible state: the *pure azimuth*.

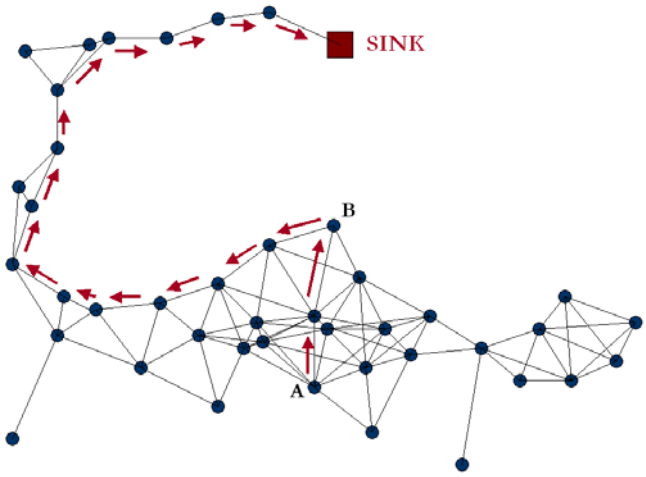


Fig. 3. The packet generated in the node A is being forwarded to the node B according to the *pure greedy* state of the routing algorithm. In the node B, algorithm switches to the *greedy/azimuth* state and the packet is forwarded to the sink.

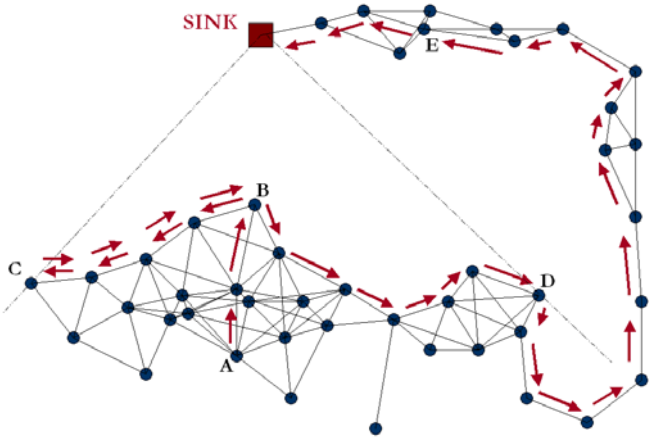


Fig. 4. The packet is being forwarded from the node B to the node C according to the *greedy/azimuth* state. In the node C, the forwarding direction is changed to the opposite one and the greedy/azimuth state is continued until the node D is reached. There, there are no good forwarders again and the routing state switches to *pure azimuth*. The node E is the first node which is closer to the sink than the best previous node (node B), so the *pure greedy* state is launched again there.

C. Pure azimuth state

In the *pure azimuth* state, the algorithm is trying to follow the borderline of the sensor network (Fig. 4). As a forwarder, a node is chosen that has the least azimuth angle relative to the direction to the previous node (Fig. 5). The position of the sink is of no significance, but the forwarding direction is considered in order to stay at the border of the network. This routing state is continued until a node is reached that is closer to the sink than it is indicated by the *least distance value*. Then, the algorithm switches to the *pure greedy* state, again.

As we mentioned at the beginning of Section II, the positions of two previous nodes are also transmitted in the packet header. This additional data is used in the *greedy/azimuth* and *pure azimuth* states to check if a potential forwarder is really located at the borderline of the network. In order to do that, the routing algorithm verifies if two vectors are not crossed (vectors A-B and C-D in Fig. 2 and vectors B-C and D-E in Fig. 5). This allows to avoid routing loops.

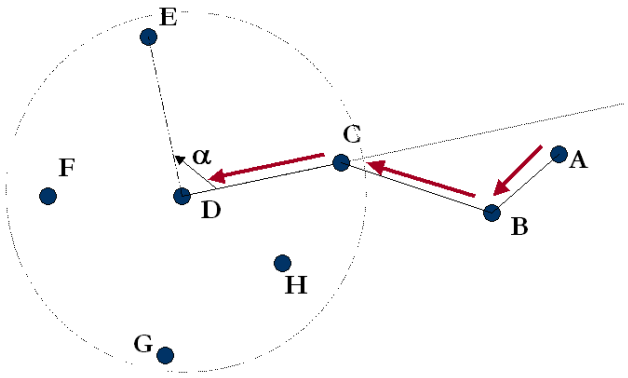


Fig. 5. According to the rules of the *pure azimuth* state, the node D is looking for a forwarder which has the least azimuth angle relative to the direction to the previous node C. The forwarding direction is clockwise, so the azimuth angle must be calculated with counter-clockwise rotation (angle α) in order to follow the borderline of the network. All the nodes in the range (E, F, G and H) are considered, but the forwarder will be the node E.

D. Mathematical background

As we have described above, all the routing decisions are always taken by the analysis of the relative positions of the actual node, its neighbours and the nearest sink. Neighbour nodes must calculate their azimuth angles and check if they are at the appropriate side of the actual forwarder according to the forwarding direction. Also, they need to check if the appropriate vectors are not crossed. Still, the suitable mathematical equations are very basic and not too demanding for sensor processors. There are explained with details in the Appendix of the paper.

E. Adverse network topologies

There exist some adverse network topologies where the presented routing algorithm fails to forward a packet to a sink. An example of such a topology is shown in Fig. 6. In the next section the probability of such a situation is discussed.

F. MAC protocol

To perform appropriately, the azimuth routing should be combined with a medium access control (MAC) protocol. This thorough analysis of MAC issues is out of scope of this paper, but we only should note that some suitable solutions have been already proposed in the open literature as BLR [5] and GeRaF protocols [6, 7]. The main idea of these protocols is to construct a smart time division multiple access scheme where the potential forwarders are retransmitting the packet in the successive time slots. The first slot is occupied by the forwarder with the best metric (closest to the sink) and its transmission prevents other potential forwarders from sending the packet again. Thus, the forwarding node is automatically chosen. While BLR and GeRaF protocols are designed for the greedy routing, they can be also generalized for the azimuth routing.

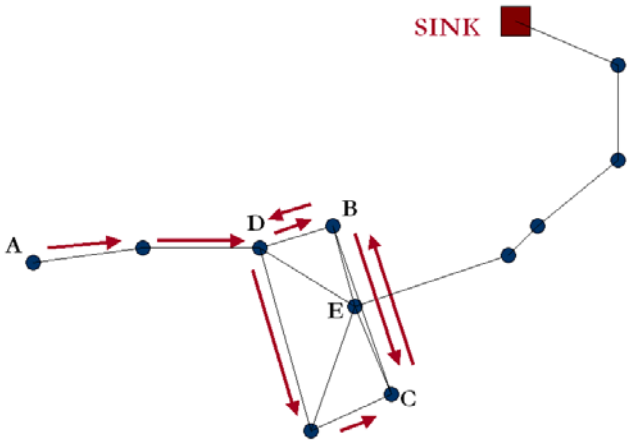


Fig. 6. An example of a network topology where the azimuth routing fails. The packet is generated in the node A and forwarded in the *pure greedy* state to the node B. There, the routing switches to the *greedy/azimuth* state and the packet is forwarded to the node C where the forwarding direction is changed and the packet is sent back until the node D. In the node D, the routing state switches to *pure azimuth* state and the packet starts to circulate in the loop. The only way to exit is the node E that never won't be reached.

III. Algorithm Performance

The proposed azimuth routing protocol was implemented in C++ and some computer simulations were performed. To characterize the connectivity between sensor nodes, we use the well-known and widely used unit disk graph model [8]. This model assumes that all the nodes have the same transmission power and equal circle range. With these assumptions, the entire network topology can be scaled up or down to obtain the sensor range equal to 1, independently of the transmission power and the path loss.

The network topology was generated randomly, with a 2-dimensional uniform distribution of the nodes and the sinks in a square area. We analysed the routing performance for the variable node density (average number of neighbours). Basically, the network consisted of 300 nodes and 3 sinks. We were changing the area where the nodes were distributed in order to obtain different average number of neighbours. In the case of low-density networks, most of the nodes were not connected, i.e. there was no path between them and any sink. The simulation parameters (average number of neighbours, percentage of nodes with successful routing) were calculated only for the connected nodes.

In Fig. 7, the performance of the azimuth routing is illustrated and it is compared with the case when only the greedy routing is applied. It should be stressed that for all cases, there is a probability that azimuth routing fails. These failure probabilities for azimuth routing are shown additionally in Fig. 8 (values from Fig. 8 added to values of upper curve from Fig. 7 sum up to 100%). The worst failure values occur for mid-density networks (about 5 neighbours in average) and do not exceed 1%. It can be explained as follows. For low-density networks, the nodes have only 1-3 neighbours in average, the network topologies are rather simple and the shortest path to the sink is very often the only path. On the other hand, when the node density is high, most of the packets can be delivered with greedy procedure only. The performance for mid-density networks is critical.

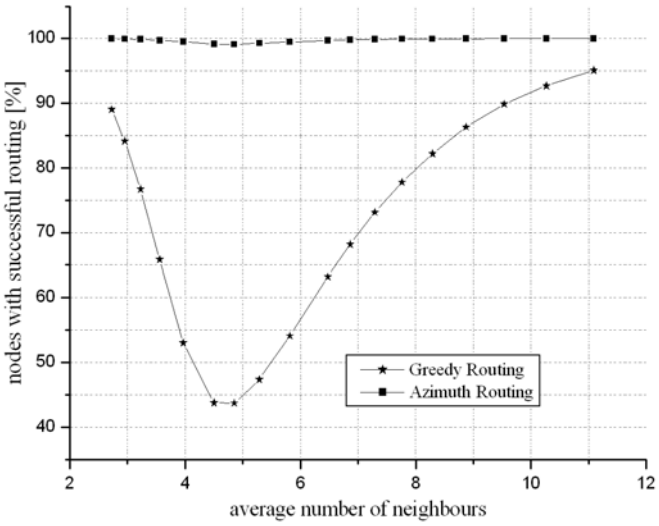


Fig. 7. The percentage of the nodes that can deliver their packets to a sink. The greedy and azimuth routing algorithms are compared.

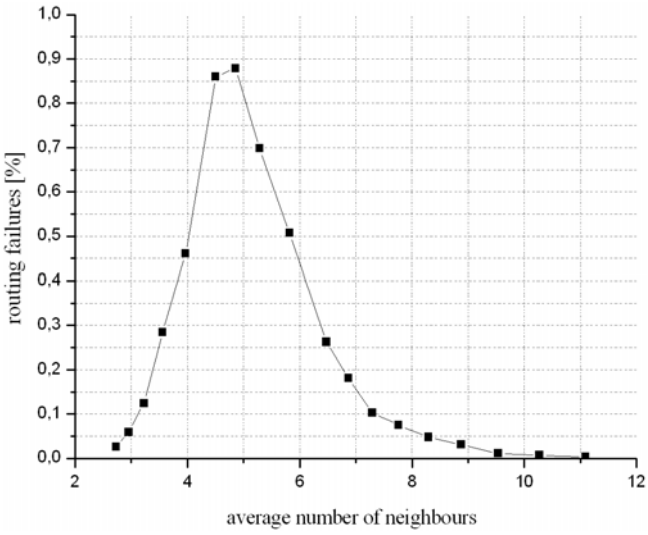


Fig. 8. The percentage of the nodes that cannot deliver their packets to a sink in azimuth routing.

We suppose the failure probabilities can be larger if wireless channel fadings are taken into account (more realistic connectivity models). Channel fadings make the network topology exemplified in Fig. 6 more probable. We plan to investigate this issue in our future work.

However, it is questionable if the issue of routing failures does matter in practice. If wireless sensors are deployed randomly or quasi-randomly (e.g. by an airplane or a helicopter) there is a significant percentage of not connected nodes (Fig. 9), unless the network density is very large. A very small percentage of routing failures seems to be not important, when 20 % of nodes are not connected. On the other hand, if the network topology is planned (not random), the adverse cases can be avoided.

As a comment to Fig. 7, we also note that the good performance of greedy routing for very small network density is misleading. If a low-density wireless network is generated randomly, the most of the nodes are not connected (Fig. 9).

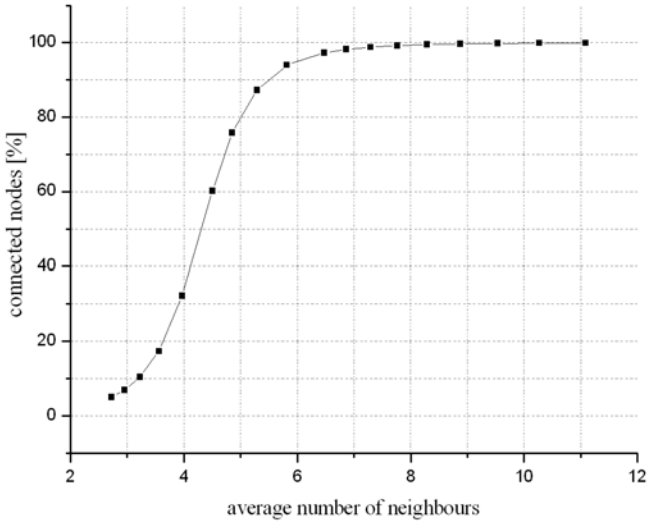


Fig. 9. The percentage of the connected nodes in the randomly deployed network.

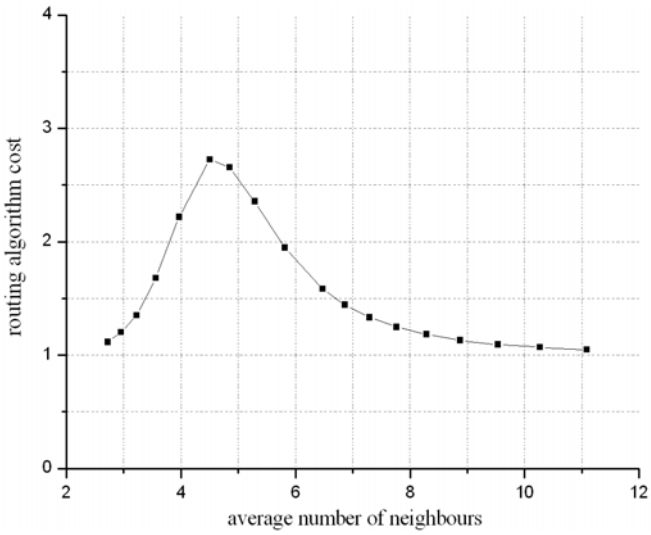


Fig. 10. The algorithm cost for the azimuth routing.

The algorithm cost of azimuth routing is shown in Fig. 10. The cost of routing is defined as the average number of hops done with the protocol to reach the sink divided by the average number of hops in the shortest possible path [4]. Obviously, the algorithm cost is largest for the mid-density networks, where greedy routing has a large probability to fail and *greedy/azimuth* and *pure azimuth* states are required. A similar curve for the *face routing* can be found in references [4, 9], also for the networks modelled with unit disk graph assumption. The algorithm cost for the face routing is a little worse: it exceeds 3 for mid-density networks, while the maximum cost of azimuth routing is 2.7. However, it is difficult to rigorously compare these two results: the authors of [4, 9] do not specify the size of the modelled network. It should be also noted that the face routing requires extra data exchange before the whole routing procedure is started (planarisation process). On the other hand, the face routing guarantees the packet delivery [4].

Finally, in Fig. 11 we illustrate the probability of azimuth routing failure as a function of the number of nodes in the network. The number of sinks is increasing proportionally with one aditional sink for each additional 100 nodes. The network density is kept constant: the average number of neighbours is equal to 5. The failure probability is increasing with the number of nodes, but it doesn't exceed 1.5 % even for 1000 nodes. It can be explained by the fact that, as the sinks are also located randomly, the average number of hops to a sink is slightly increasing. Thus, it results in the larger probability of adverse network topologies.

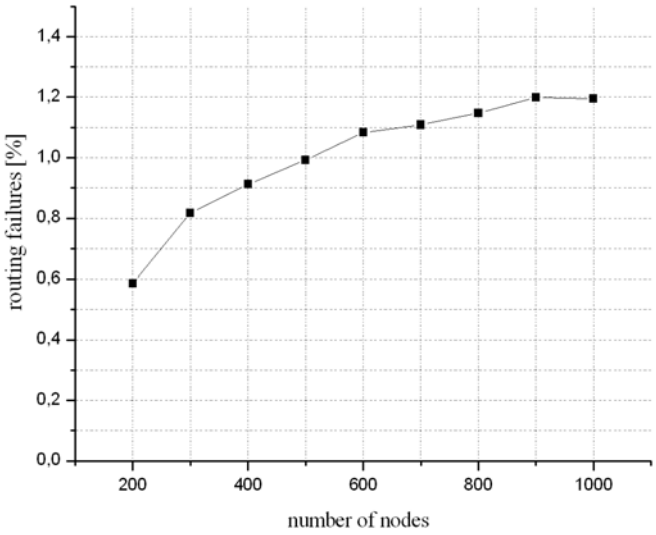


Fig. 11. The percentage of the nodes that cannot deliver their packets to a sink in azimuth routing as a function of the number of nodes in the network.

IV. Conclusions

The new algorithm called *azimuth routing* was proposed. It was a geo-routing protocol that did not require any knowledge about the network topology. The algorithm performance was validated through the computer simulations and good results were obtained. As future work, we plan to verify the algorithm in a more diverse propagation environments and to integrate it with a medium access control protocol.

Appendix

To make routing decisions, sensors must perform simple calculations concerning their positions and the positions of nearby nodes. In particular, they need to:

a. calculate the azimuth angles,
b. check at which side of a line they are located,
c. check if two vectors are crossed.

A. Azimuth angles

In the *greedy/azimuth* state, the best forwarder is chosen according to the azimuth angle. In order to do that, we propose that each potential forwarder node calculates the following metric:

$$m = \frac{(x_f - x_0)(x_s - x_0) + (y_f - y_0)(y_s - y_0)}{\sqrt{(x_f - x_0)^2 + (y_f - y_0)^2}}, \qquad (1)$$

where (x_f, y_f), (x_s, y_s) and (x_0, y_0) are the coordinates of the potential forwarder (neighbour node), the sink and the transmitting node, respectively. According to the definition of the dot product:

$$m = \cos\alpha \cdot \sqrt{(x_s - x_0)^2 + (y_s - y_0)^2}\ , \qquad (2)$$

where α is the azimuth angle of the potential forwarder relative to the direction to the sink. The square root in (2) equals to the distance from the sink to the transmitting node and is, of course, constant in the metrics of all the potential forwarders. Thus, the neighbour node with the largest metric *m* has the least azimuth angle.

In the *pure azimuth* state, it is also necessary to calculate the azimuth angles. The procedure is analogous to the one presented above.

B. Line side

In *greedy/azimuth* state, each potential forwarder should also check if it is on the appropriate side of the line connecting the transmitting node and the sink – to fulfil the condition regarding the forwarding direction. It can be performed by calculating the following expression:

$$c = (x_s - x_0)(y_f - y_0) - (x_f - x_0)(y_s - y_0)\,. \qquad (3)$$

According to the definition of the cross product, the sign of the above expression gives the information about the position of the potential forwarder relative to the line connecting the transmitting node and the sink.

C. Two crossing vectors

In order to avoid routing loops in *greedy/azimuth* and *pure azimuth* states, the potential forwarders check if two vectors are crossed, as it was explained in Section II.C. Again, the equation (3) is useful. If, e.g. the vectors AB and CD should be checked (see Fig. 2), it is enough to verify if the points A and B are at the same side of the line connecting point C and D.

ACKNOWLEDGEMENTS

This work was supported by Polish project grants 179/N-COST/2008/0 and DWM/MOB17/II/2008. It was also supported by a Spanish grant TEC2007-67966-01/TCM (CON-PARTE-1) and developed in the framework of "Programa de Ayudas a Grupos de Exelencia de la Region de Murcia, Fundacion Seneca".

REFERENCES

[1] I. Stojmenovic (ed.), *Handbook of Sensor Networks. Algorithms and Architectures*. John Wiley & Sons, 2005.

[2] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, 2005.

[3] E. Kranakis, H. Singh and J. Urrutia, "Compass routing on geometric networks," In *Proc. of 11th Canadian Conference on Computational Geometry*, pp. 51–54, Vancouver, Canada, August 1999.

[4] F. Kuhn, R. Wattenhofer and A. Zollinger "An Algorithmic Approach to Geographic Routing in Ad Hoc and Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 11, pp. 51-62, February 2008.

[5] M. Heissenbuttel, T. Braun, T. Bernoulli and M. Walchli, "BLR: beacon-less routing algorithm for mobile ad hoc networks," *Computer Communications*, vol. 27, no. 11, pp.1076-1086, July 2004.

[6] M. Zorzi and R. R. Rao, "Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Multihop Performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 337-348, October-December 2003.

[7] M. Zorzi and R. R. Rao, "Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Energy and Latency Performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 349-365, October-December 2003.

[8] I. Stojmenovic, "Simulations in Wireless Sensor and Ad Hoc Networks: Matching and Advancing Models, Metrics, and Solutions," *IEEE Communications Magazine*, vol. 46, no. 12, pp. 102-107, December 2008.

[9] F. Kuhn, R. Wattenhofer, Y. Zhang and A. Zollinger "Geometric Ad-Hoc Routing: Of Theory and Practice," In *Proc. of 22nd ACM Int. Symp. on the Principles of Distributed Computing*, pp. 63-72, Boston, USA, 2003.

Localización en WLAN utilizando distribuciones de probabilidad con reducción de cómputo por trilateralización

Miguel A. Quintana, David Sánchez, Domingo Marrero, Juan Luis Navarro(1)
Grupo de Arquitectura y Concurrencia
Departamento de Ingeniería Telemática, Departamento de Señales y Comunicaciones(1)
Universidad de Las Palmas de Gran Canaria
Campus Universitario de Tafira, 35017, Las Palmas de Gran Canaria
{mquintana, dsanchez, dmarrero}@dit.ulpgc.es, jnavarro@dsc.ulpgc.es

***Resumen*- La localización en redes inalámbricas es un área de investigación importante debido a que proporciona información de interés que puede ser utilizada por otras aplicaciones para tomar mejores decisiones y mejorar su rendimiento. Los métodos de localización deberían ser diseñados para obtener resultados con la mayor precisión posible y con un coste computacional bajo. Sin embargo, esta tarea no es fácil porque normalmente se requiere de muchos cálculos para alcanzar una precisión elevada. En este artículo, se presenta un método de localización sobre redes de infraestructura 802.11 basado en distribuciones de probabilidad que utiliza la técnica de trilateralización para determinar un punto de partida donde comenzar la búsqueda de la solución. Una vez se obtiene el punto de partida, el área de búsqueda estará limitada a las localizaciones más cercanas a este punto.**

***Palabras Clave*- Localización, distribuciones de probabilidad, trilateralización, nivel de señal recibido**

I. INTRODUCCIÓN

Hoy en día, el desarrollo de dispositivos de computación móvil y los avances en las tecnologías inalámbricas permiten utilizar ordenadores portátiles en muchas situaciones de la vida cotidiana. Uno de los desafíos abiertos de la computación móvil es determinar la localización de usuarios [1]. En este sentido, la computación basada en localización ha hecho posibles aplicaciones con capacidad para tomar decisiones en función de su ubicación y modificar sus prestaciones y funcionalidad en concordancia a su entorno [2]. Las herramientas de localización se pueden utilizar en el desarrollo de aplicaciones relacionadas con la navegación, la seguridad, la salud o el entretenimiento.

Durante los últimos años muchos autores han dedicado sus esfuerzos en resolver el problema de la localización en las redes IEEE 802.11. En este contexto, la localización en interiores todavía se considera un problema fundamental en la computación móvil donde no está resuelto con claridad el requisito de obtener una precisión elevada [3].

Tradicionalmente, se han utilizado diferentes modelos para determinar la localización, tales como el ángulo de llegada (AOA), diferencia de tiempo de llegada (TDOA) y la potencia de la señal recibida (RSS) de la señal de RF. Esta última técnica es una de las modalidades más utilizadas debido a que la información utilizada para determinar la localización se obtiene a través de la propia interfaz de red y, por tanto, no son necesarios sensores adicionales. Esta información se utiliza en algunas técnicas como la trilateralización, centroides ponderados, etc. Entre las técnicas más utilizadas se encuentran aquellas que se basan en un análisis del RSS en cada posición. La utilización de esta técnica se debe principalmente a que los efectos de la multitrayectoria y los efectos de propagación como la reflexión, la difracción y la dispersión causan una estimación inexacta del ángulo de llegada y de la diferencia de tiempo de llegada [4]. Por esa razón, la mayoría de los trabajos de investigación utilizan un modelo basado en la potencia de la señal. Este modelo se construye en dos fases. En la primera fase, normalmente denominada fase de entrenamiento, se construye una base de datos con la potencia de señal recibida en cada posición. En la segunda fase, denominada fase de clasificación, partiendo de una muestra de RSS y un sistema de clasificación que utiliza la información almacenada en la base de datos se determina la posición. La mayoría de los trabajos difieren en la última fase.

En [5] se utiliza un método de clasificación de datos basado en la máquina de vectores soporte (SVM) con resultados aceptables. Su principal inconveniente es que requiere un tiempo adicional para el entrenamiento de la SVM. En [6] se utilizan dos métodos para determinar la ubicación en entornos interiores y exteriores: un método basado en el algoritmo del vecino más cercano y un método basado en que la propagación de RSS disminuye logarítmicamente con la distancia. Los autores indican que el primer método produce resultados más coherentes en los ambientes exteriores. Sin embargo, su exactitud puede verse afectado si hay muchos puntos con datos similares. Esta situación puede darse en ambientes interiores, debido a la propagación multitrayecto. El modelo de propagación es más preciso en ambientes con visión directa pero menos preciso en el resto de casos.

Por otro lado, también son utilizados métodos probabilísticos para determinar la posición. En [7] los autores utilizan distribuciones de probabilidad combinadas con propagación de señales para tener en cuenta las características de absorción y reflexión de los diversos obstáculos. La

principal ventaja de este método es la utilización de un único punto de acceso. Sin embargo, los autores reconocen que la máxima precisión que se consigue es determinar la posición dentro de una habitación en un edificio. En [8] se utilizan también distribuciones de probabilidad, teniendo en cuenta la alta correlación entre las muestras de un mismo punto de acceso, combinadas con una técnica de agrupamiento para reducir el número de cálculos. En función del número de puntos de acceso y la posición de estos, el tamaño de los grupos pueden ser grandes, y por tanto, la reducción de cálculo no es tan efectiva. Esto último ha de tenerse en cuenta si la localización se realiza en dispositivos de mano donde la potencia del procesador no es elevada y la energía almacenada en la batería es limitada. Con el fin de reducir el área de búsqueda, en este trabajo de investigación se propone utilizar la técnica de trilateralización para determinar un punto de partida de la búsqueda. De esta manera, la zona de búsqueda se limita a los lugares más cercanos a este punto. Por tanto, en este artículo se presenta un método de localización basado en distribuciones de probabilidad que utiliza la técnica trilateralización para reducir el área de búsqueda.

El resto del documento está organizado de la siguiente forma. En la sección 2, se describe la técnica trilateralización utilizada para determinar el punto de partida de la búsqueda. A continuación, en la sección 3, se explica el método de distribuciones de probabilidad utilizadas para deducir la localización del dispositivo. En la sección 4, se muestran algunos resultados experimentales. Por último, se resumen las conclusiones y se indican las líneas futuras a seguir.

II. TÉCNICA DE TRILATERALIZACIÓN

La trilateralización es una técnica por la puede ser determinada la localización de un dispositivo estimando la distancia entre el dispositivo y los puntos de acceso de los cuales se recibe señal. La trilateralización suele utilizarse en las comunicaciones celulares para determinar la posición geográfica de un usuario. En las redes IEEE 802.11 basadas en infraestructura los puntos de acceso suelen estar en una posición fija permitiendo la comunicación de los dispositivos en un área de cobertura. Si son conocidas la posición de tres puntos de acceso (vértices A, B, C del triángulo en la Fig. 1) y las distancias $\overline{DA}$, $\overline{DB}$ y $\overline{DC}$, entonces puede utilizarse el método de trilateralización para deducir la posición D. Esto se realiza mediante la búsqueda de la intersección de las tres de circunferencias cuyos centros son los vértices A, B y C. Por tanto, dadas las coordenadas de cada punto de acceso (x_i, y_i) y las distancias desde el dispositivo portátil a cada uno de ellos (d_i), se obtiene la posición de un dispositivo mediante el sistema no lineal que se indica a continuación.

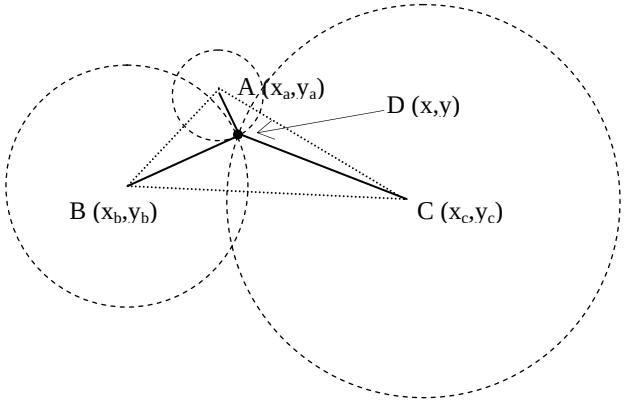


Figura 1. Técnica de trilateralización

$$(x - x_i)^2 + (y - y_i)^2 = d_i^2 \qquad (1)$$
$$i = a, b, ..., n$$

Con el fin de obtener la posición de un dispositivo es preciso conocer la ubicación de al menos tres puntos de acceso. Si son conocidos sólo dos puntos de acceso, entonces el sistema devuelve dos posibles soluciones, y si sólo está accesible un punto de acceso, las posibles posiciones del dispositivo se encuentran en la circunferencia con un radio igual a la distancia al punto de acceso.

Para resolver (1) es necesario determinar la distancia desde el dispositivo a cada punto de acceso. Para hacerlo, nos basamos en que la propagación de la señal disminuye logarítmicamente con la distancia, la cual se puede expresar como [9]:

$$P = S + 10 \times n \times \log(d) + \sum L_w \qquad (2)$$

donde P es la pérdida del canal en dB desde el punto de acceso al dispositivo (potencia transmitida menos potencia recibida), S es la de pérdida del canal en dB a 1 metro del punto de acceso, n es el factor de atenuación que depende de las características de propagación del entorno, d es la distancia en metros entre el transmisor y el receptor, y L_w representa las pérdidas de penetración cuando hay paredes entre el transmisor y el receptor. Por tanto, dado un valor de RSS, es necesario conocer el factor de atenuación y las pérdidas en el canal para determinar la distancia d (3). Estos valores deben ser estimados de forma empírica para cada punto de acceso debido a que dependen del entorno de propagación y, por regla general, se utilizan transmisores con diferentes características y ubicados en diferentes lugares.

$$d = \log^{-1}\left(\frac{P - S - \sum L_w}{10 \times n}\right) \qquad (3)$$

Para el cálculo del factor de atenuación, n, se deben tomar numerosas muestras de RSS en cada posible ubicación por cada transmisor y sólo cuando existe visión directa entre transmisor y receptor. A continuación, se estima un factor de atenuación para cada punto de acceso, y para cada posible posición, por aplicación de la ecuación (4). Por último, se calcula un factor de atenuación promedio equivalente por cada punto de acceso.

$$n_{di} = \left(\frac{P - S}{10 \times \log(d_i)}\right) \qquad (4)$$

Por otra parte, si en algunas posiciones no hay visión directa entre transmisor y receptor, ha de estimarse la pérdida por penetración. Estos valores son obtenidos por la diferencia entre la RSS cuando el transmisor está en visión directa y cuando hay una o varias paredes entre ellos. Una vez estimados todos los parámetros y partiendo de un conjunto dado de valores RSS, obtenidos de *k* puntos de acceso, recogidos en una determinada posición, se calcula la distancia del dispositivo a cada uno de los punto de acceso mediante (3). A continuación, una vez conocida la posición de los puntos de acceso y las distancias a estos dispositivos, se resuelve (1).

III. Determinación de la Localización utilizando Distribuciones de Probabilidad

En esta sección se describe el método basado en las distribuciones de probabilidad para estimar la localización. En concreto, se describe el proceso realizado en las fases de entrenamiento y clasificación, y las ventajas de usar trilateralización para reducir el área de búsqueda.

A. *Fase de entrenamiento*

Durante esta fase se construye una base de datos donde para cada punto de posible localización se almacenan las muestras recibidas de cada punto de acceso. Para realizar esta tarea, primero, tenemos que definir los puntos posibles de la localización en el entorno, y a continuación, se debe realizar un proceso de muestreo de RSS en cada uno de estos puntos. Para cada posición se almacena un conjunto de *n* muestras (r_1,…, r_k) recibido desde *k* puntos de acceso (AP_1,…, AP_k), donde r_i es el valor de la potencia de señal recibida desde el AP_i.

Partiendo de estos valores, la probabilidad $P(r_i/l)$ de que un determinado nivel de señal r_i esté en una localización determinada *l* puede ser estimada usando el histograma normalizado del punto de acceso AP_i para esa localización. Los valores de la probabilidad para cada punto de acceso y localización se almacenan en la base de datos. En la Fig. 2 y en la Fig. 3 se muestra un ejemplo de la potencia de señal normalizada obtenida a diez y veinte metros de un punto de acceso, respectivamente. Como se puede apreciar en las figuras, la propagación de la señal varía con la distancia. En [8] y [10] se indica que la distribución de RSS se puede aproximar a una distribución normal. Sin embargo, en nuestros experimentos hemos detectado que la distribución depende del entorno y de cómo se realice el muestreo de datos. Por ejemplo, en nuestros experimentos hemos recogido muestras en cuatro orientaciones diferentes en cada posición, es decir, se recopilaron datos con el dispositivo orientado hacia el norte, este, sur y oeste. En la Fig. 4 y en la Fig 5. se muestra un ejemplo de 200 datos recogidos en cada localización, donde cada cincuenta muestras el dispositivo se cambia de orientación (giro de noventa grados). Como se puede apreciar, el valor medio de RSS cambia en cada dirección y en cada localización. Por tanto, podemos decir que la distribución de RSS se puede aproximar a una mezcla de varias distribuciones normales (Fig. 2 y Fig. 3), con pequeñas variaciones espaciales que afectan a la distribución de la señal.

El tamaño del conjunto de muestras obtenida en cada localización y cómo éstas se recogen puede repercutir en la exactitud del sistema. Por tanto, para obtener una buena estimación de la distribución de probabilidad es necesario construir una base de datos con un conjunto amplio de muestras recogidas en diferentes orientaciones y en cada posición.

B. *Fase de clasificación*

En esta fase se determina la localización del dispositivo partiendo de un conjunto de muestras de valores de RSS obtenidos desde varios puntos de acceso. En el sistema desarrollado en este artículo, la localización del dispositivo viene determinada por aquella posición que, dado un conjunto de valores de RSS, presenta la probabilidad más alta. Para determinar la probabilidad en cada punto utilizamos el teorema de Bayes.

Dado una muestra de los valores de RSS R=(r_1,…, r_k) obtenidos de *k* puntos de acceso, el objetivo es encontrar una localización *l* tal que $P(l/R)$ sea la mayor. Aplicando el teorema de Bayes, $P(l/R)$ puede ser expresado como:

$$P\left(l/R\right) = \frac{P\left(R/l\right) \times P(l)}{P(R)} \qquad (5)$$

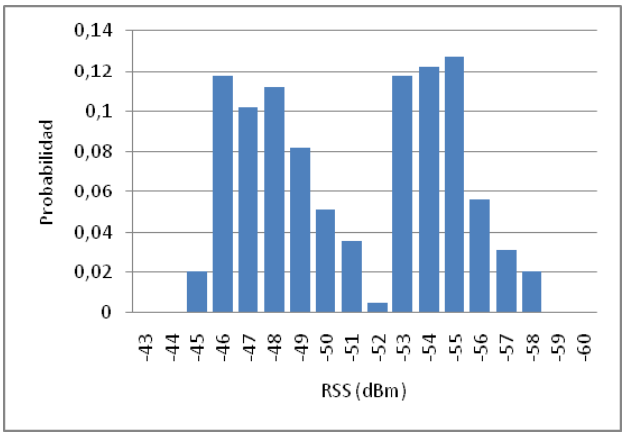


Figura 2. Histograma a 10 metros de distancia desde el punto de acceso.

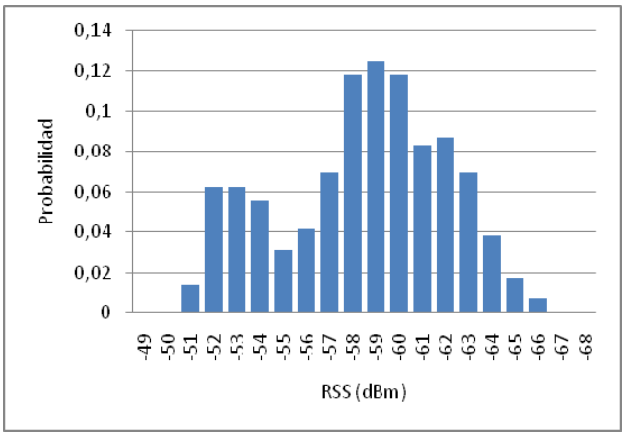


Figura 3. Histograma a 20 metros de distancia desde el punto de acceso.

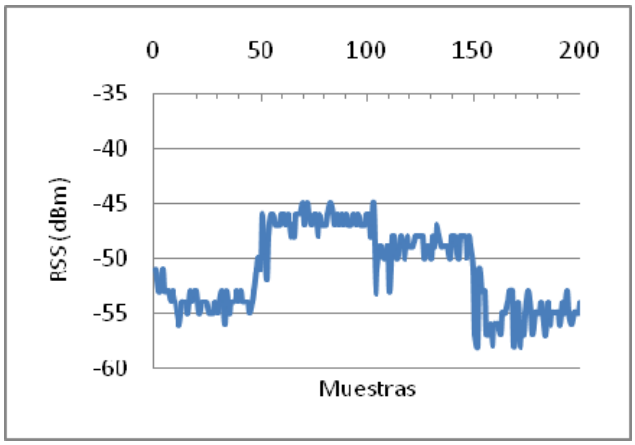


Figura 4. RSS recogidas en cuatro orientaciones diferentes y a 10 metros de distancia del punto de acceso.

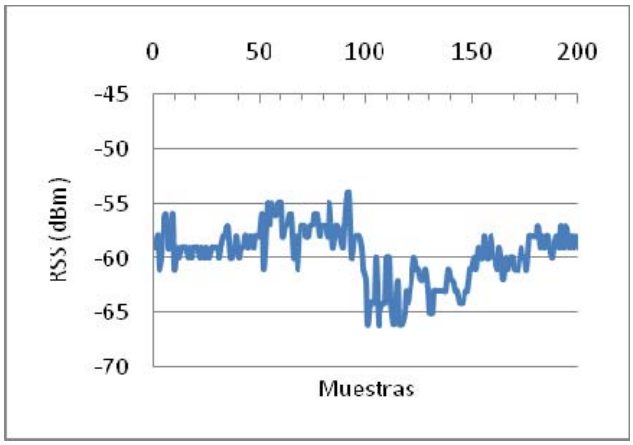


Figura 5. RSS recogidas en cuatro orientaciones diferentes y a 20 metros de distancia del punto de acceso.

Dado que se busca la localización donde se maximiza la expresión (5) y no su valor exacto, y además, teniendo en cuenta que el valor de *P(R)* no varía en las diferentes evaluaciones de la expresión, la ecuación anterior se puede reformular como:

$$P\left(l/R\right) = P\left(R/l\right) \times P(l) \qquad (6)$$

Por otra parte, al diseñarse el sistema para determinar la localización, sin memoria sobre su posición anterior, la probabilidad de que un usuario esté en una determinada localización *P(l)* es igual para todas las posibles localizaciones. Por tanto, este factor influye de la misma forma en todos *P(l/R)* y, así se puede extraer de (6). Luego, la ecuación (6) puede ser definida como:

$$P\left(l/R\right) = P\left(R/l\right) \qquad (7)$$

Si asumimos que los puntos de acceso no operan en canales solapados, los valores de RSS recibidos de estos se pueden considerar como independientes. Por tanto, *P(R/l)* puede ser calculado como sigue:

$$P\left(R/l\right) = P\left(r1/l\right) \times \cdots \times P\left(rk/l\right) = \prod_{i=1}^{k} P\left(ri/l\right) \qquad (8)$$

donde $P(r_i/l)$ se calcula a partir de las muestras recogidas en cada localización en la fase de entrenamiento

C. Reducción del área de búsqueda

Tal como hemos mencionado en la sección anterior, la determinación de la localización basada en distribuciones de probabilidad se estima para aquella localización donde *P(R/l)* es máxima. Por tanto, este valor debe ser calculado para todas las posibles localizaciones que estén dentro del área de cobertura de los puntos de acceso, aplicando (8). El número de posibles localizaciones puede ser excesivamente grande. Este valor depende del entorno, de la posición de los puntos de acceso y del tamaño de cada posible localización. Por ejemplo, si en una determinada localización los valores recogidos de RSS pertenecen a un único punto de acceso, el espacio de búsqueda se extiende al área de la cobertura de ese punto de acceso. Sin embargo, si los paquetes de señalización se reciben de dos o más puntos de acceso el espacio de la búsqueda se reduce a la intersección de todas las áreas de la cobertura. La Fig.6 muestra un ejemplo de la intersección del área de la cobertura de dos puntos de acceso. Para realizar la búsqueda en el área de la intersección es necesaria una clasificación previa en la fase de entrenamiento para determinar las posibles áreas de localización en base a un determinado conjunto de valores de RSS. En cualquier caso, implica un coste computacional, y por tanto, consumo de energía. Este aspecto se debe tomar en consideración si la determinación de la localización se realiza en dispositivos de mano.

Para reducir el número de operaciones, en este artículo se propone utilizar el método de trilateralización, de manera que la intersección de tres circunferencias nos dará el punto de partida para comenzar la búsqueda. Debido a que el factor equivalente de atenuación usado para solucionar (3) se calcula por la aproximación de datos empíricos, y que las condiciones del entorno puede cambiar de manera aleatoria, los resultados obtenidos mediante (1) pueden ser distintos a la solución correcta. Por tanto, la búsqueda se debe realizar sobre un nuevo círculo de radio *s* centrado en la solución (1). El valor de este radio *s* del área de búsqueda es un compromiso entre la precisión del sistema y el coste computacional. Si los valores de RSS se recogen a partir de dos puntos de acceso, la intersección de las dos circunferencias nos dará dos puntos de partida de búsqueda. En la Fig. 7 se indican las áreas de búsqueda, zonas rojas, cuyos centros son las intersecciones de las circunferencias con radio igual a las distancias de los puntos de acceso al dispositivo. Finalmente y en el caso peor, si sólo está disponible un punto de acceso, la búsqueda se realiza a lo largo del anillo formado al desplazar el círculo de radio *s* a lo largo de la circunferencia de radio igual a la distancia *d*.

En la tabla 1 y 2 se muestra la reducción del área de búsqueda en función del número de puntos de acceso cuando se utiliza la intersección de áreas de cobertura y la cobertura del punto de acceso del que se recibe mayor potencia, respectivamente. En estas tablas, el valor de *r* se corresponde con el radio del área de cobertura de un punto de acceso (suponemos un mismo radio de cobertura para todos los puntos de acceso), *a* es el área de una posible localización o unidad de partición del espacio de localización, *d* es la distancia del punto de acceso al dispositivo desde donde se

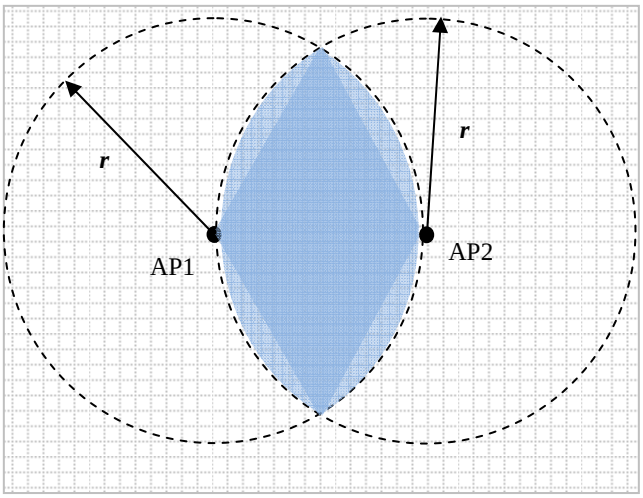


Figura 6. Intersección del área de cobertura.

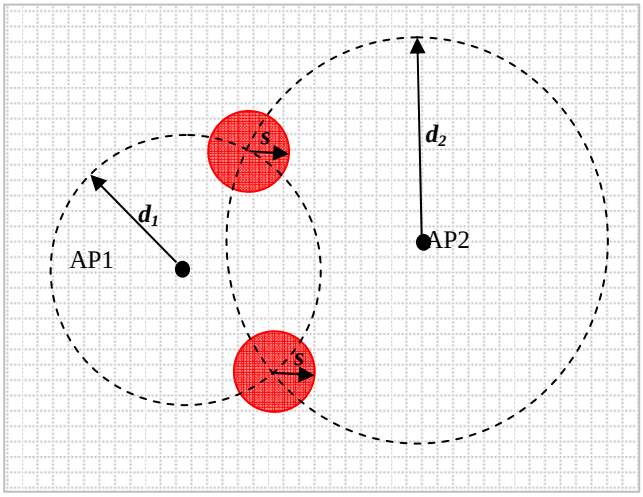


Figura 7. Reducción del espacio de búsqueda.

recogen los valores de RSS, y *s* es el radio del círculo donde se hace la búsqueda (círculo rojo en la Fig. 7). Este valor siempre será menor que el radio *r* del área de la cobertura con una relación aproximada de 10 a 1. El valor de la distancia *d* depende de la posición del usuario, pero siempre será menor o igual a *r*.

TABLA I. REDUCCIÓN DEL ÁREA DE LA BÚSQUEDA CUANDO SE UTILIZA INTERSECCIÓN DE COBERTURAS

Available Access Points	***Normal***	***Trilateration***
1	$\pi r^2 / a$	$4 \pi d s / a$
2	$(2\pi/3 - \sqrt{3}/2) r^2 / a$	$2 \pi s^2 / a$
3	$(AP1 \cap AP2 \cap AP3) / a$	$\pi s^2 / a$

TABLA II. REDUCCIÓN DEL ÁREA DE LA BÚSQUEDA CUANDO SE UTILIZA COBERTURA DEL PUNTO DE ACCESO DE MAYOR POTENCIA RECIBIDA

Available Access Points	***Normal***	***Trilateration***
1	$\pi r_i^2 / a$	$2 \pi r s / a$
2	$\pi r_i^2 / a$	$2 \pi s^2 / a$
3	$\pi r_i^2 / a$	$\pi s^2 / a$

IV. RESULTADOS

Las pruebas experimentales se han realizado en la segunda planta del Departamento de Ingeniería Telemática de la Universidad de Las Palmas de Gran Canaria. Esta planta tiene una dimensión aproximada de 63 metros de largo por 10 metros de ancho. En la Fig. 8 se muestran la disposición de la planta y la posición de tres puntos de acceso (indicados con un triángulo). Nuestros experimentos fueron realizados solamente en el pasillo, el cual tiene aproximadamente 2 metros de ancho.

En la fase de entrenamiento de nuestro sistema se ha construido una base de datos con muestras tomadas a lo largo del pasillo. En concreto, se han definido 126 celdas de aproximadamente 1 m^2. En cada celda se tomaron 200 muestras de cada punto de acceso, y cada 50 muestras el dispositivo fue cambiado de orientación (giro de noventa grados). Las medidas se realizaron durante varios días, en diferentes horarios y siempre existiendo visión directa entre el transmisor y el receptor. A partir de estos datos se obtiene el histograma por cada celda y punto de acceso. Una vez obtenidas todas las muestras, el siguiente paso fue calcular el factor medio de atenuación referente a cada punto de acceso. Para ello, y sabiendo la pérdida en el canal a 1 metro de distancia de cada punto de acceso y la potencia de transmisión de cada uno de éstos, se aplicó la ecuación (4) para calcular el factor de atenuación correspondiente a cada muestra. De estos valores, se obtiene el valor medio en cada posición. La desviación típica entre todas las posiciones es baja. Por último, se estima el valor medio de todas las posiciones para cada punto de acceso. Los valores obtenidos se muestran en la tabla 3.

Debido a que los puntos de acceso están situados en el pasillo, el área de cobertura de cada uno de ellos abarca todo el pasillo, es decir, en cada celda se reciben paquetes de señalización de todos los puntos de acceso. Dada esta situación, no fueron estimadas las pérdidas de penetración debido a las paredes, por lo que no se utilizó en (3).

Los experimentos realizados se enfocaron para evaluar la precisión y la carga computacional del sistema.

TABLA III. FACTOR DE ATENUACIÓN PARA CADA PUNTO DE ACCESO

WireLess_GAC_A	1.65
Telematica	1.53
WaveLAN	1.88

A. *Precisión del sistema*

Para evaluar la precisión del sistema de localización, por distribuciones de probabilidad, se situó el dispositivo en diferentes posiciones a lo largo del pasillo. En cada posición y en diferentes instantes de tiempo se invocó el método varias veces mientras que el dispositivo portátil permanecía estático, evaluando el total de celdas posibles. La tabla 4 muestra el porcentaje medio de probabilidad de éxito en diferentes posiciones, es decir, indica el porcentaje de veces que el método de localización por probabilidad da como posición correcta la posición que se encuentra a *x* metros de la situación real, siendo *x* igual a 0, 1, 2, 3 y mayor o igual a 4. Como se puede apreciar, el método de localización tiene, en

Figura 8. Plano de la planta donde se realizaron las pruebas.

el peor de los casos, un 49% de probabilidad de estimar la posición correcta, una precisión media cercana al 90% dentro del radio de tres metros.

TABLA IV. PORCENTAJES DEÉXITO

Posición (m)	*0 m*	*1 m*	*2 m*	*3 m*	*>= 4m*
5	71%	12%	8%	1%	8%
15	54%	23%	7%	4%	12%
25	55%	19%	13%	3%	10%
35	51%	15%	11%	7%	16%
45	59%	15%	13%	4%	9%
55	49%	19%	17%	5%	10%

B. Coste computacional

El coste computacional del sistema propuesto depende del punto de partida estimado con el método de trilateralización y la longitud del radio *s* del círculo con centro en dicho punto. De la tabla 4, podemos deducir que el valor óptimo de *s* está en tres metros debido a que representa un éxito cercano al 90%. En este trabajo se ha evaluado el coste computacional como el número de celdas a explorar para encontrar una posición donde la probabilidad es máxima. En nuestro entorno de pruebas, un valor de *s* igual a 3 representa catorce celdas a explorar. En el peor de los casos y si no se utiliza el método del trilateralización se deberán explorar las 126 celdas para encontrar la celda donde la probabilidad es máxima, por tanto el sistema de localización tiene una carga computacional del 10% respecto al total, es decir, una reducción del 90%.

Algunos autores [8] utilizan agrupamiento para clasificar las diversas áreas de búsqueda. En nuestro caso, puesto que los paquetes de señalización se reciben desde tres puntos de acceso en todas las celdas, solamente se pueden distinguir tres áreas de búsqueda. Cada una de estas áreas se identifica por el punto de acceso más cercano, señal recibida con mayor potencia. Por ello, en el mejor de los casos suponiendo una agrupación en tres zonas idénticas, éstas tendrían aproximadamente el mismo número de las celdas, 42. Por tanto, el método que presentamos en este artículo tendría una carga computacional del 31% respecto al agrupamiento, es decir, una reducción del 69%.

V. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se ha presentado un sistema de localización en redes IEEE 802.11 utilizando distribuciones de probabilidad con una precisión aceptable, 90% de éxito con un error máximo de tres metros. La mejor característica de este trabajo es su coste computacional bajo debido a la reducción del área de búsqueda. La carga computacional se reduce al 69% en el peor de los casos. Esta reducción se obtiene al realizar la búsqueda a partir de un punto estimado al aplicar la técnica de trilateralización. Por tanto, el sistema de localización consume pocos recursos, y puede ser tomado en consideración para su implementación en dispositivos de mano.

Por otro lado, en los experimentos realizados se ha utilizado la misma interfaz de red para construir la base de datos como para realizar las pruebas. Esto se debe a que las medidas y el rango de los valores de RSS dependen de la interfaz de red. Debido a que los fabricantes utilizan diferentes interfaces, una nueva línea de trabajo se enfoca en diseñar una nueva técnica de localización basada en la diferencia entre los niveles de RSS para eliminar la dependencia con el tipo de tarjeta de red. Además, también estamos estudiando la posibilidad implementar filtros de Kalman para reducir las variaciones espaciales y temporales de RSS.

Además de las líneas de trabajo abiertas que hemos comentado en el párrafo anterior estamos interesados en obtener mejores resultados mediante el método de trilateralización y reducir así el radio de búsqueda. Debemos observar que el cálculo del punto de partida puede verse afectado en gran medida por la aproximación a un constante del factor de atenuación (4). Proponemos ajustar este valor para cada uno de los puntos de acceso en la fase de entrenamiento a curvas que dependan del valor *di* y sean factible de ser utilizadas mediante una formulación similar a (3).

AGRADECIMIENTOS

Este trabajo ha sido subvencionado por la Universidad de Las Palmas de Gran Canaria a través del proyecto precompetitivo de investigación ULPGC 07/030, y el Departamento de Ingeniería Telemática.

REFERENCIAS

[1] A. Zomaya, "Mobile Computing: Opportunities for Parallel Algorithms Research", 15th IEEE IPDPS, pp. 144–147, 2002.

[2] R. Want and B. Schilit, "Expanding the Horizons of Location-Aware Computing", IEEE Computer, pp. 31-34, August 2001.

[3] L. Zong, D. Kotz, R. Jain and X. He, "Evaluating next-cell predictors with extensive Wi-Fi mobility data", IEEE Transactions on Mobile Computing, vol 5, 12, pp. 1633-1649, December 2006.

[4] P. Bahl and V.N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system", IEEE Conference on Computer Communications (INFOCOM'00), pp. 775-784, March 2000.

[5] C. Wu, L. Fu, F. Lian, "WLAN Location Determination in e-Home via Support Vector Classification", IEEE Conference on Networking, Sensing & Control, Taiwan, pp. 1026-1031, 2004.

[6] M. Emery and M. Denko, "IEEE 802.11 WLAN based real-time location tracking in indoor and outdoor environments". Canadian Conference on Electrical and Computer Engineering, pp: 1062-1065, 2007.

[7] G. V. Zaruba, M. Huber, F. A. Kamangar and I. Chlamtac "Indoor location tracking using RSSI readings from a single Wi-Fi access point". Wireless Network, Springer, vol. 13, pp: 221-235, 2007.

[8] M. Youssef and A. Agrawala, "The Horus WLAN location determination system", IEEE International Conference on Mobile Systems, Applications, and Services, 2005.

[9] A. Santamaría, F. López-Hernández, *Wireless LAN Systems*, Artech House, 1994.

[10] K. Kaemarungsi, "Distribution of WLAN received signal strength indication for indoor location determination", International Symposium on Wireless Pervasive Computing, 2006.

Evaluación de mecanismos de priorización en 802.11p con VHDL

Juan Bautista Tomás Gabarrón, Esteban Egea López, Joan García Haro
Departamento de Tecnologías de la Información y las Comunicaciones,
Universidad Politécnica de Cartagena, Plaza del Hospital 1, 30202 Cartagena
{juanba.tomas, esteban.egea, joang.haro}@upct.es

***Resumen*—El estándar para comunicaciones inalámbricas en redes vehiculares, IEEE 802.11p, ofrecerá mecanismos de priorización de la información para permitir una diferenciación de la calidad de servicio entre aplicaciones. Debido a las condiciones especiales a las que están sujetas las redes vehiculares, es importante realizar una sintonización de los parámetros que configuran los procedimientos de diferenciación de servicios. En este artículo se presenta el desarrollo de un modelo en VHDL (*VHSIC HDL - Very High Speed Integrated Circuit Hardware Description Language*) que permite evaluar cómo reaccionan las interfaces de salida en un nodo que implementa IEEE 802.11p frente a distintas configuraciones de los parámetros de ajuste mencionados. El análisis de la distribución de ancho de banda entre interfaces permitirá discernir cómo se administra este reparto en nodos equipados con IEEE 802.11p.**

***Palabras Clave*—Redes Vehiculares, FPGA, VANET, WAVE, 802.11p.**

I. Introducción

En los últimos diez años hemos asistido a un avance importante en el uso de redes inalámbricas para el acceso a Internet y para la formación de redes de ámbito local y personal (LANs y PANs) [1]. El estándar IEEE 802.11 [2] ha constituido desde sus inicios la solución más atractiva para la interconexión de terminales inalámbricos y así lo demuestra el gran número de dispositivos comercializados hasta la fecha que implementan esta tecnología [3]. A lo largo de todo este tiempo este estándar ha experimentado ciertas modificaciones que lo hacen más idóneo para las características especiales que reune cada entorno de operación. Es el caso de IEEE 802.11a [2], que introduce modificaciones en la capa física mediante la utilización de *OFDM* (multiplexación por división de frecuencias ortogonales) para incrementar las tasas de transmisión, o algunas más recientes como 802.11e [2], que ofrece capacidades de calidad de servicio (QoS) mediante la diferenciación de servicios con interfaces de distinta prioridad.

En la actualidad están emergiendo nuevos mecanismos de aprovechamiento de los recursos que ofrecen las redes inalámbricas. Es el caso de las redes *ad-hoc* móviles (MANET). Este tipo de redes se caracteriza principalmente por estar constituido por nodos con capacidades de movilidad, lo que implica que los procesos de transmisión y recepción se vean afectados por el tiempo y la posición espacial de los mismos. Un caso específico de estas redes son las *VANET* (redes *ad-hoc* vehiculares), que intercambian información de interés o bien entre vehículos o entre vehículos e infraestructura, y proporcionan así a los pasajeros prestaciones avanzadas en materia de seguridad, control del tráfico rodado, ocio y acceso a información. Con este propósito se ha desarrollado recientemente una modificación profunda del estándar de comunicaciones inalámbricas IEEE 802.11 [2] que pretende regular la transmisión de información en entornos vehiculares. Este estándar adopta de forma específica la denominación IEEE 802.11p [4]. Debido a las características particulares de las transmisiones en entornos de tráfico vehicular, el estándar implementa una serie de mecanismos que lo hacen más idóneo que el 802.11 original. Además, IEEE 802.11p hereda los procedimientos de diferenciación de servicios que ya contemplaba la extensión 802.11e mediante la creación de una serie de interfaces que permiten administrar el servicio de los paquetes según la prioridad que tengan asignada.

Es propósito de este artículo investigar sobre el mecanismo de priorización que IEEE 802.11p ofrece, mediante la implementación mediante lenguaje VHDL de las operaciones que el estándar detalla, para un dispositivo hardware programable *FPGA Xilinx Spartan 3E* [5]. El objetivo principal consiste en el análisis preliminar de la distribución de ancho de banda que la estructura de interfaces en un nodo equipado con 802.11p realiza, según los valores asignados a los parámetros que determinan la prioridad para cada interfaz de servicio. El resto del artículo se estructura de la siguiente forma: la sección 2 contempla una descripción detallada de los aspectos más importantes que describen la funcionalidad del estándar IEEE 802.11p para redes vehiculares; la sección 3 detalla algunos trabajos relacionados con el desarrollo VHDL para la evaluación de protocolos de comunicación inalámbrica; la sección 4 expone detalladamente el modelo generado con el lenguaje de especificación hardware VHDL [6]; en la sección 5 se exponen y discuten los resultados obtenidos. Finalmente en la sección 6 se concluye el estudio.

II. Arquitectura WAVE

WAVE (*Wireless Access in Vehicular Environments*) constituye la arquitectura de protocolos que administra las capas de nivel de red, enlace, acceso al medio y física para las comunicaciones en VANETs [4] (ver Fig. 1). IEEE 802.11p [4] representa la propuesta realizada por el IEEE dentro de la arquitectura WAVE para la regulación de los mecanismos de acceso al medio en redes *ad-hoc* vehiculares (VANET). Durante la redacción de este artículo, IEEE 802.11p todavía se encuentra en fase de finalización, y se estima Abril de 2009 como posible fecha de publicación oficial del estándar por parte del *IEEE 802.11 Task Group p*.

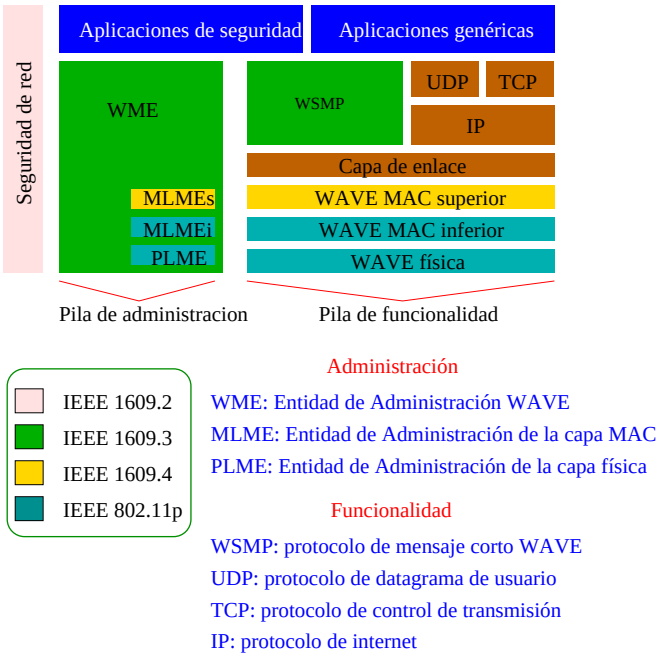


Fig. 1. Arquitectura de protocolos en WAVE (*Wireless Access in Vehicular Environments)*

A. DSRC (Dedicated Short Range Communications)

La historia de la arquitectura WAVE se remonta a 1999 cuando la FCC (*Federal Communications Commission*) estadounidense estableció un espectro de 75 MHz en la banda de los 5.9 GHz (banda de los ITS, *Intelligent Transportation Systems*) para albergar de manera exclusiva las tecnologías emergentes de radiocomunicaciones que tendrían lugar en las redes vehiculares de nueva generación. DSRC [7] (*Dedicated Short Range Communications*) es el nombre que adopta el espectro en esta banda, y se estructura según siete canales de 10 MHz cada uno (ver Fig. 2).

A diferencia de las comunicaciones en la banda de 2 GHz, cuya utilización está asociada a las redes WiFi y Bluetooth [8], la banda de los ITS requiere licencia para su uso.

El nivel físico propuesto por el DSRC se estructura en supertramas de duración predeterminada de 100 ms. Cada intervalo asociado a una supertrama se divide asimismo en dos períodos cada uno de los cuales se dedica a un aspecto particular de la comunicación (véase la Fig. 3). El primero de ellos es el CCH (canal de control), cuya duración por defecto es de 50 ms y que se encarga del envío de información importante, normalmente relacionada con la seguridad en la conducción, así como de la administración en la formación de grupos de entidades vehiculares según una aplicación determinada (subredes de ámbito específico). El segundo es el SCH (canal de servicio), compuesto por la multiplexación temporal cada 100 ms de canales que operan a distintas frecuencias dentro de la banda de los ITS. SCH permite la transmisión de información relacionada con seguridad, ocio y administración remota a través de paquetes IP (protocolo de Internet).

La diferencia fundamental entre CCH y SCH radica en el

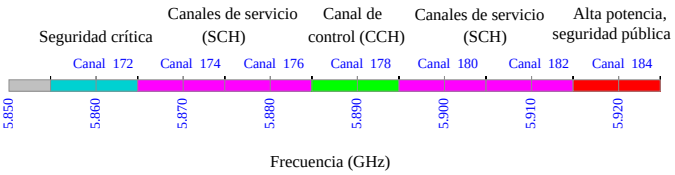


Fig. 2. Estructura del espectro en la banda de los ITS (*Intelligent Transportation Systems)*

hecho de que el primero no puede usar IP para la transmisión de paquetes. Para ello recurre a un protocolo de propósito específico que opera al mismo nivel que IP y que representa el acrónimo WSMP (*WAVE Short Message Protocol*) [4]. WSMP toma consideración de las características especiales que definen a los entornos de tránsito vehicular y reduce sustancialmente la carga de los paquetes (funcionalidad reducida de las cabeceras) para mejorar el caudal en las transmisiones.

B. IEEE 1609/802.11p

Los servicios de red de la arquitectura WAVE proporcionan capacidades de distribución de datos entre dispositivos WAVE y administración entre capas de la arquitectura de red. Esta última se compone de los estándares IEEE Std 1609.1TM-2006 (presentación), IEEE Std 1609.2TM-2006 (servicios de seguridad), IEEE Std 1609.3TM-2006 (servicios de red, WSMP), IEEE Std 1609.4-2006 y IEEE 802.11p (capas de acceso al medio). Por ser el eje central de nuestro desarrollo, nos centraremos en IEEE 802.11p, esquema fundamental de acceso al medio.

El esquema básico de acceso que contempla WAVE se denomina IEEE 802.11p y se basa en acceso múltiple por escucha de portadora (CSMA) [9]. Dicho mecanismo se denomina EDCA (*Enhanced Distributed Channel Access)* [4], y se utiliza tanto para el acceso en comunicaciones V2I (vehículo a infraestructura) como V2V (vehículo a vehículo), en esquemas de acceso cliente-servidor y distribuidos. Deriva de la DCF (Función de Coordinación Distribuida) especificada para el estándar original 802.11 [2], pero integra la diferenciación de servicios mediante la definición de cuatro interfaces virtuales de acceso que implementan funcionalidades de *backoff* para la transmisión de las tramas. Cada interfaz se configura con una serie de parámetros relativos al tamaño máximo de ventana de *backoff* así como intervalos de espaciado AIFS (*arbitrary inter-frame space*) que vendrán determinados por la clase de acceso que dicha interfaz tenga asignada. Esto permite establecer distintas calidades de servicio para los numerosos tipos de

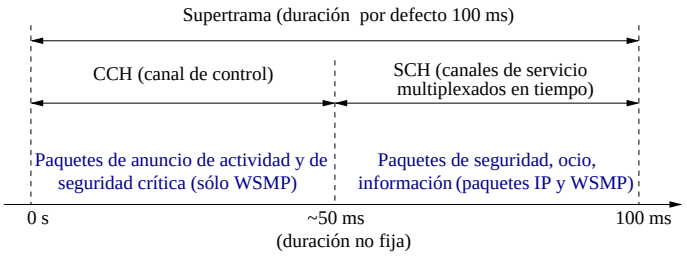


Fig. 3. Estructura particular de la supertrama en WAVE

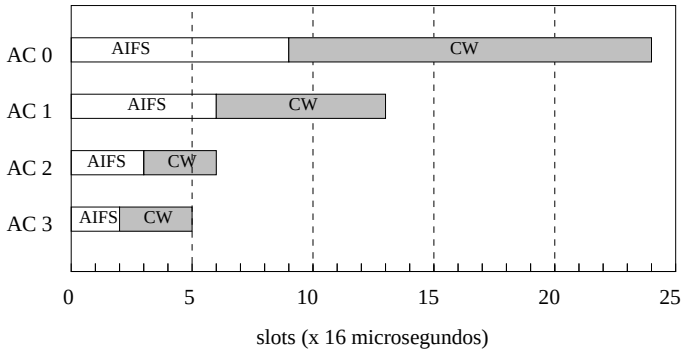


Fig. 4. Configuración de parámetros de *backoff* en IEEE 802.11p

aplicaciones que se prevé servir.

EDCA se basa en el servicio de paquetes de datos según la prioridad asignada a la aplicación que los genera. Para ello, cada estación mantiene configuraciones independientes para cada una de las cuatro interfaces definidas lo cual permite establecer distintas calidades de servicio (QoS) según el tipo de aplicación al que quiera darse servicio. Cada interfaz opera con mecanismos de acceso según escucha de portadora, es decir, cada estación virtual monitorizará el medio durante un intervalo determinado de tiempo (AIFS) a partir del cual ejecutará un proceso de *backoff* decremental en el que escogerá un número de slots aleatorio entre el tamaño máximo de ventana de *backoff* (*CWmax*) y el valor mínimo (*CWmin*) (ver Fig. 4). En caso de identificarse una colisión entre paquetes en el momento de transmitir, se reproducirá una nueva fase de *backoff* en la que los parámetros de anchura máxima y mínima de ventana se modificarán para reducir la probabilidad de colisión en intentos de acceso sucesivos (ver Tabla 1). Si se alcanza el número máximo de intentos posibles, se descarta el paquete y se reiniciarán los valores de CW a los iniciales para el nuevo paquete que se desee enviar al medio.

III. Trabajos relacionados

Algunos grupos de investigación se han interesado en la implementación física con VHDL de los protocolos asociados a redes inalámbricas basadas en 802.11. Todavía no existe ningún trabajo específico que se base en el estándar IEEE 802.11p, aunque sí existen numerosos estudios que evalúan los distintos estándares o modificaciones surgidos a partir del original (IEEE 802.11).

En primer lugar puede citarse la referencia [10], cuyos autores diseñaron la capa de acceso al medio (*enhanced MAC*) para la provisión de servicios multimedia sobre redes inalámbricas con 802.11. El prototipo diseñado soporta la funcionalidad de dos tipos de capa física: OFDM (*Orthogonal Frequency Domain Multiplexing*), basada en la utilización de frecuencias ortogonales con portadoras a 5 GHz para la modulación de la información a transmitir, y DSSS (*Direct Sequence Spread Spectrum*), que utiliza mecanismos de espectro expandido, con portadoras centradas a la frecuencia de 2 GHz. La implementación permite cambiar de un mecanismo de capa física a otro de manera realmente sencilla.

En [11] se presenta el diseño de un prototipo basado en OFDM para la capa física que opera según el estándar 802.11a, muy similar en capa física a IEEE 802.11p. En dicho trabajo se realiza un estudio detallado de los mecanismos necesarios para la implementación de la capa física sobre una plataforma de diseño FPGA de Xilinx. En el sistema se incluye lógica de sincronización para la detección de paquetes y el establecimiento de los tiempos de operación. Se presenta además una metodología de trabajo realmente útil de cara al desarrollo de prototipos asociados a la comunicación en redes inalámbricas 802.11.

El estudio llevado a cabo en [12] introduce un prototipo desarrollado para implementar las comunicaciones a través del estándar 802.11a, aunque se centra en la construcción de una arquitectura *single chip* que permite el ahorro en costes de producción y en consumo de potencia. Para ello se pretende reducir la complejidad en el desarrollo y optimizar al máximo la integración de componentes.

En [13] se describe una arquitectura que implementa el protocolo 802.11a en una estructura *on-chip*. Se presenta el flujo completo de diseño del dispositivo, desde la simulación del protocolo mediante SDL (*Software Description Language*) [14] hasta la integración de componentes en la estructura *on-chip*. Asímismo se explotan los recursos de hardware dedicado para una mejora sustancial del rendimiento en tareas críticas.

IV. Modelo del sistema

Como hemos podido observar, el funcionamiento del estándar de comunicaciones inalámbricas para redes vehiculares se basa en la configuración de una serie de parámetros que determinan la operación de cada interfaz virtual en los nodos equipados con 802.11p. De una manera más detallada, los parámetros que configuran la funcionalidad son los siguientes:

- Duración del intervalo arbitrario entre tramas (AIFS). Este valor es fijo para cada interfaz, y se debe completar siempre la duración de dicho intervalo antes de comenzar con la siguiente etapa de *backoff*.
- Ancho de la ventana de *backoff*
  - Tamaño máximo de la ventana (*CWmax*). Determina el máximo valor obtenible de hallar el número aleatorio previo a la etapa de *backoff* decremental.
  - Tamaño mínimo de la ventana (*CWmin*). Establece el valor mínimo seleccionable cuando se obtiene el valor aleatorio antes de comenzar el *backoff*.
- Longitud del intervalo de transmisión de paquetes para cada interfaz, o período de oportunidad para la transmisión (TXOP, *transmission oportunity*).

El objetivo que se persigue en este estudio es evaluar cómo reaccionan las interfaces frente a distintos valores de estos

tres parámetros. Para ello se define un modelo que emula la operación real de una estación equipada con 802.11p, y se efectúa una serie de pruebas en las que analizamos cómo opera el sistema en las situaciones que veremos en la siguiente sección.

La estructura generada con VHDL para el presente trabajo se basa en la operación de dos módulos principales donde cada uno de ellos desempeña una función particular pero estrechamente relacionada con la del otro. En primer lugar, existe un módulo denominado *backoffModule* el cual se encarga de llevar a cabo los procedimientos relacionados con las esperas en los intervalos de *backoff* para cada una de las cuatro interfaces que componen el sistema. Este módulo contiene asimismo una estructura generadora de números aleatorios que permite introducir la aleatoriedad requerida por el proceso de *backoff*. En segundo lugar, el módulo *control* tiene la función de controlar los procedimientos de *backoff* que se realizan en el primer módulo según las señales que éste provee. A continuación detallaremos la arquitectura generada en cada nivel de abstracción.

A. Estructura de primer nivel

La estructura de mayor nivel puede visualizarse en la Fig. 5. Esta implementación se realiza para facilitar las pruebas posteriores con la tecnología *FPGA Spartan 3E*. Como se puede observar, en esta estructura existe un total de siete entradas, cuya función se analiza a continuación:

- **reset.** Esta entrada está conectada directamente a un generador de números aleatorios que se encargará de dotar de aleatoriedad al procedimiento de *backoff*. Sólo se utiliza esta entrada para poner en funcionamiento el generador, ya que no es necesaria su reinicialización posterior una vez que se comienza con el proceso de generación.
- **enable.** Este puerto está conectado al generador de números aleatorios, y determina si se encuentra en funcionamiento o no.
- **aifs.** Este bus de entrada establece cuáles son los tiempos de espera entre tramas (AIFS) antes del procedimiento de *backoff*. Concretamente, es uno de los parámetros de configuración que usaremos para evaluar el comportamiento de la arquitectura. Para las cuatro interfaces existentes se destinan 3 bits que contendrán la información relativa al tamaño del período AIFS particular para cada una de ellas.
- **CWLength.** Este puerto de entrada de tipo bus proporciona la información relativa al tamaño máximo que puede tomar la ventana de contienda de cada interfaz. Éste es otro de los parámetros que modificaremos en las pruebas posteriores.
- **txop.** Este bus determina cuál es la duración máxima de cada interfaz para la transmisión de información al medio por cada intento de transmisión.
- **sload.** Puerto de entrada que sirve para hacer una precarga de los contadores al inicio de las simulaciones.
- **clk.** Señal de reloj del dispositivo.

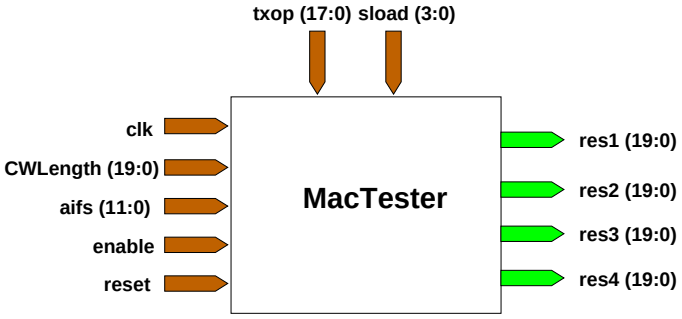


Fig. 5. Estructura de mayor nivel de la implementación en VHDL (en paréntesis se especifica el rango en bits de los puertos)

Las salidas de este dispositivo informan sobre cuál es el número de ciclos que se asigna a cada interfaz durante un intervalo determinado. En total son cuatro salidas, cada una de las cuales es un bus que proporcionará un valor relativo al tiempo que cada interfaz accede al medio. Cada salida recibe el nombre de *resX*, donde X $(X = 1...4)$ se refiere al número de interfaz correspondiente, mientras que cada una posee un tamaño de 20 bits.

B. Estructuras de segundo nivel

En el segundo nivel (nivel de abstracción siguiente al anterior en orden descendente encontramos una estructura compuesta por *MaController* y *AccessCounter* (ver Fig. 6). El módulo más importante es el primero, que en definitiva guarda toda la lógica que implementa nuestro sistema. El segundo módulo es simplemente una estructura que contabiliza el número de ciclos que cualquiera de las cuatro salidas del bus del primer módulo permanece a 1 lógico (ciclo de tiempo asignado a una interfaz en particular).

C. Estructuras de tercer nivel

Esta estructura detalla los módulos que componen el dispositivo *MaController* descrito en el subapartado anterior. *MaController* implementa cuatro módulos de *backoff* cada uno correspondiente a sendas interfaces dentro de la arquitectura IEEE 802.11p (véase Fig. 7). Estos módulos están también conectados a un módulo de control que se encarga de administrar las señales necesarias para el procedimiento de *backoff*, como por ejemplo, la señal de ocupación del medio (*chBusy*) o la señal de aviso sobre fin de transmisión (*endTx*). A continuación detallamos estas entradas y otras que son en definitiva, las que caracterizan

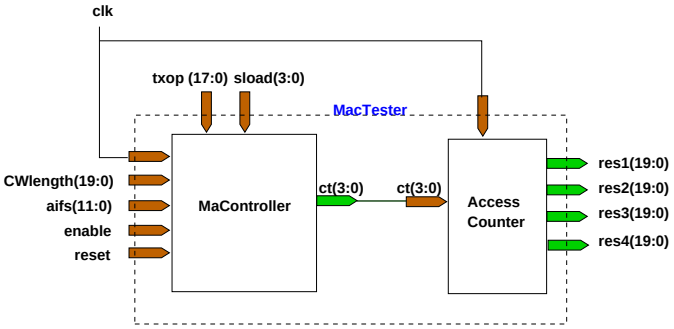


Fig. 6. Estructuras de segundo nivel de la implementación en VHDL (en paréntesis se especifica el rango en bits de los puertos)

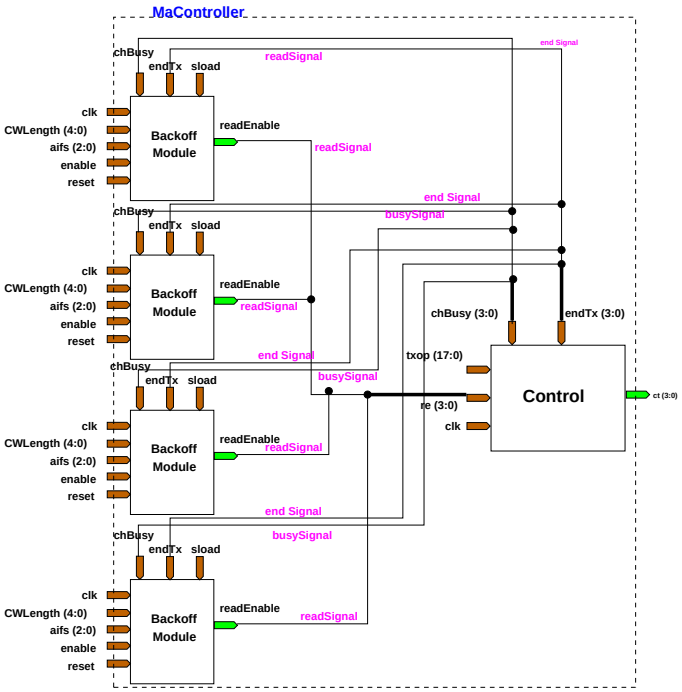


Fig. 7. Estructuras de tercer nivel de la implementación en VHDL (en paréntesis se especifica el rango en bits de los puertos)

este nivel de abstracción:

- **chBusy.** Esta entrada procede del elemento de control y tiene la misión de informar a cualquier módulo de *backoff* la situación de disponibilidad del canal: libre (0) u ocupado (1). En caso de que esté ocupado, el módulo de *backoff*, o bien esperará para iniciar el proceso de *backoff* hasta que quede nuevamente libre, o bien si ya había comenzado uno lo posterga hasta el siguiente momento en el que se pueda monitorizar el canal como desocupado.
- **readEnable.** Ésta es la única salida de cada uno de los módulos de *backoff*, y determina en un momento en particular si ha terminado el proceso de *backoff* para una interfaz concreta (1) o todavía se encuentra en estado de decremento (0). La señal de cada módulo de *backoff* irá a parar al módulo de *Control*, para que sea éste quien administre qué estación virtual tiene el derecho a transmitir en un momento determinado según el esquema de prioridades que se haya asignado previamente.
- **endTx.** Esta señal indica a aquella interfaz que haya comenzado previamente una transmisión, si dicha emisión de información ha terminado ya (mediante un pulso). Se utilizará para administrar las pausas de los módulos de *backoff* y las señales que gobiernan el estado actual del medio (libre u ocupado).
- **txop.** En el momento de capturar el medio para transmisión, cada estación tendrá derecho a emitir paquetes al medio durante un intervalo de tiempo determinado. Este intervalo de tiempo viene dado por los parámetros de configuración que se hayan asignado a la estructura por medio de este puerto de entrada.

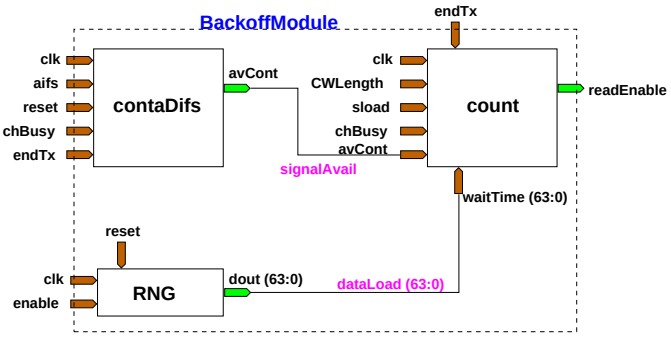


Fig. 8. Estructuras de cuarto nivel de la implementación en VHDL (en paréntesis se especifica el rango en bits de los puertos)

D. Estructuras de cuarto nivel

Los módulos que todavía permiten ahondar más en su implementación interna son específicamente los de *backoff*. Concretamente se basan en la utilización de dos contadores y un generador de números pseudoaleatorios (véase Fig. 8). En primer lugar, un contador realiza la cuenta atrás correspondiente al período AIFS que cada interfaz en particular debe realizar cuando hay información para transmitir y además se monitoriza el medio libre. Una vez se determina que el medio ha quedado libre tras una transmisión, el contador AIFS informa al contador de *backoff* que ya puede comenzar el procedimiento decremental del mismo nombre. Al mismo tiempo, cuando comienza el período AIFS, el generador de números pseudoaleatorios proporciona un valor de 32 bits que se usa para inicializar el contador de *backoff* a un número de slots aleatorio (dentro de los márgenes impuestos por el valor procedente del puerto AIFS) que se usará en el momento en el que se comience con el procedimiento.

V. Experimentación

En esta sección se incluye una serie de pruebas experimentales con variaciones en los valores propios de los parámetros de *backoff* para comprobar cómo se establece la prioridad entre las distintas interfaces que componen el sistema. Todas las pruebas se realizarán a un tiempo de funcionamiento de 10 ms, intervalo más que suficiente para asegurar la operación del sistema en régimen permanente, determinado por la utilización de una frecuencia de reloj de 100 ns que permite efectuar los experimentos con aproximadamente 100000 ciclos de reloj totales. Debido a la pequeña cantidad de ciclos de reloj asociados a los parámetros AIFS y TXOPs, que determinan los tiempos de ocupación, frente al número total de ciclos mencionados en emulación, es obvio el alcance del régimen permanente.

Cada subsección de esta parte se corresponderá con una configuración particular de parámetros, se analizarán los resultados obtenidos en la gráfica de la Figura 9 y se terminará con una justificación de los mismos. Concretamente, en dicha Figura se represetan los resultados de cada experimento (marcados con un color distinto para cada uno de los experimentos) obtenidos para cada una de las interfaces; en el eje de abcisas cada interfaz se identifica con un número de 1 a 4. Estos resultados se corresponden con el porcentaje de ocupación de cada interfaz respecto al total (número total de ciclos

Tabla I
VALORES DE CONFIGURACIÓN PARA CADA EXPERIMENTO

	Exp. 1	**Exp. 2**	**Exp. 3**	**Exp. 4**	**Exp. 5**
CWMax1	7 ciclos	7 ciclos	7 ciclos	7 ciclos	7 ciclos
CWMax2	7 ciclos	7 ciclos	5 ciclos	7 ciclos	5 ciclos
CWMax3	7 ciclos	7 ciclos	4 ciclos	7 ciclos	4 ciclos
CWMax4	7 ciclos	7 ciclos	3 ciclos	7 ciclos	3 ciclos
txop1	7 ciclos	7 ciclos	7 ciclos	3 ciclos	3 ciclos
txop2	7 ciclos	7 ciclos	7 ciclos	7 ciclos	7 ciclos
txop3	7 ciclos	7 ciclos	7 ciclos	15 ciclos	15 ciclos
txop4	7 ciclos	7 ciclos	7 ciclos	31 ciclos	31 ciclos
AIFS1	3 ciclos	5 ciclos	3 ciclos	3 ciclos	2 ciclos
AIFS2	3 ciclos	4 ciclos	3 ciclos	3 ciclos	3 ciclos
AIFS3	3 ciclos	3 ciclos	3 ciclos	3 ciclos	4 ciclos
AIFS4	3 ciclos	2 ciclos	3 ciclos	3 ciclos	5 ciclos

correspondientes a 10 ms). El carácter del tráfico utilizado se corresponde con el de una fuente de tipo *greedy* (siempre existe información para transmitir).

A. Experimento 1

Esta primera prueba tiene el objeto de observar cómo reaccionan las interfaces cuando el valor de los parámetros de *backoff* para cada interfaz es idéntico, es decir, tanto los anchos de ventana, como las duraciones de los intervalos AIFS y TXOPs serán iguales para cada clase de prioridad. El resultado debería ser unos tiempos de acceso al medio estadísticamente muy similares para cada clase. Si observamos la configuración establecida para este caso (ver Tabla I, columna Exp. 1) obtenemos unos resultados como los mostrados en la Fig. 9 para la tonalidad negra. Se puede observar que la distribución de ancho de banda es equitativa, con la salvedad de que aquella clase de acceso de mayor prioridad obtendrá una tasa ligeramente mayor de accesos que el resto por el hecho de que en aquellos casos en los que se produzcan colisiones entre interfaces dentro de la misma estación, la de mayor prioridad (en este caso, la 4) será quien finalmente obtenga el derecho para transmitir.

B. Experimento 2

En esta nueva prueba se mantienen fijos los valores asociados a la anchura de las ventanas y a las oportunidades de transmisión, como puede observarse en la Tabla I para la columna Exp. 2. Los únicos parámetros que se diferencian se corresponden con los espaciados de tiempo entre tramas AIFS). Según los resultados correspondientes (Fig. 9, representados por la tonalidad amarilla), es posible ver de manera muy evidente que aquellas interfaces con menores tiempos de AIFS obtienen un mayor número de accesos al medio, lo que prueba la distinción entre clases de prioridad. Además, en este caso se obtiene una tasa de utilización del canal algo menor que la del caso anterior, sobre todo debido a la limitación de las clases de menor prioridad a las que se les asigna menores anchos de banda.

C. Experimento 3

En este caso se modifican los anchos máximos de ventana admisibles para cada interfaz, manteniéndose el orden de

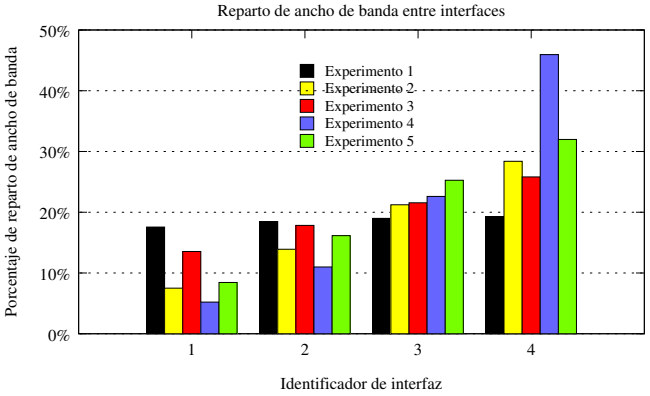


Fig. 9. Relación de ancho de banda asignado para cada interfaz en cada experimento

la asignación de prioridades como en el caso anterior (ver Tabla I, columna Exp. 3). También pueden observarse en este caso las distintas prestaciones en cuanto a la adquisición del medio para cada clase de prioridad según el máximo ancho de ventana de *backoff* (ver Fig. 9, tonalidad roja). Sin embargo, es necesario resaltar que en este caso la distinción entre porcentajes de distribución de ancho de banda no es tan alta entre interfaces como en el Experimento 2 debido al hecho de que en el procedimiento de *backoff* se obtienen números aleatorios entre 0 y el tamaño máximo de la ventana (*CWMax*), mientras que en la Prueba anterior siempre debía contabilizarse el número total de ciclos correspondientes al AIFS de la clase que quiere establecer la transmisión. Como consecuencia, no se produce la misma ”injusticia” de reparto del ancho de banda entre las interfaces de menor prioridad como en el caso previo, y por este motivo la utilización del medio también se incrementa considerablemente, rondando el 80% (resultado de sumar los porcentajes asociados a cada interfaz).

D. Experimento 4

Según la configuración expuesta para este caso (ver Tabla I, columna Exp. 4), en el que se varían los tiempos de adquisición del medio para cada clase de prioridad, en esta prueba deberían obtenerse los tiempos de adquisición del medio más diferentes para cada interfaz, ya que la magnitud de la configuración de cada *txop* es también mayor. Puede observarse en la Fig. 9, tonalidad morada, que en comparación con los casos anteriores, para mayores valores de *txop* se obtienen diferencias aun más notables en la distribución de los accesos según la clase de prioridad. Asimismo, a pesar del limitado ancho de banda asignado a las clases de menor prioridad, en este caso como es lógico la utilización del medio es mayor porque se reducen los tiempos dedicados al procedimiento de *backoff* y AIFS, con una utilización de casi el 85%.

E. Experimento 5

En este caso se realiza una configuración de parámetros que engloba todos los cambios efectuados en las pruebas anteriores (ver Tabla I, columna Exp. 5). En la Fig. 9, tonalidad verde, podemos observar que, aunque los AIFS están configurados para dar más prioridad a las clases con orden decreciente (AIFS1 tiene más prioridad que AIFS2 y así

sucesivamente), el parámetro que más influencia tiene sobre el resultado final es precisamente el correspondiente a las duraciones de tiempo asignadas a cada clase de acceso en particular (*txop*). Por ello puede inferirse que *txop* determina en gran medida cuál va a ser la asignación de ancho de banda a cada entidad, lo que sin embargo supondrá una mayor limitación de ancho de banda en las clases de menor prioridad, pero a la vez la utilización del medio también será más grande.

VI. Conclusiones y trabajo futuro

En este artículo se describe la implementación de un único esquema de interfaces bajo 802.11p en VHDL. Dicho estudio pretende ser un trabajo preliminar para el diseño de un dispositivo sobre hardware reconfigurable que permita el establecimiento de comunicaciones entre dispositivos equipados con el estándar IEEE 1609/802.11p (WAVE). Las pruebas a las que ha sido sometida la arquitectura diseñada se basan en analizar el comportamiento entre clases de servicio para evaluar la dependencia de las distribuciones de ancho de banda respecto de la configuración de parámetros. Debido a que la pretensión de este primer experimento era simplemente evaluar las prestaciones internas de un dispositivo aislado, sin entrar en el análisis de la funcionalidad referida a multitud de equipos en comunicación simultánea, los resultados, aunque prometedores, se refieren a condiciones de mejor caso. Esto resta realismo a los experimentos, ya que en una situación real se dispondría de una gran cantidad de nodos en comunicación simultánea que, por otra parte, supondría una degradación notable en la comunicación y asignaciones de ancho de banda por interfaz bastante menores a las que hemos relatado. Sin embargo, en este primer trabajo se ha podido entrever algunos aspectos clave para el funcionamiento del protocolo que ayudan a comprender su funcionalidad de manera detallada.

Por otra parte, a partir de los experimentos se ha podido poner de relieve la enorme influencia que tienen los parámetros de configuración sobre las prestaciones de diferenciación de servicios ofrecidas por las interfaces de los dispositivos 802.11p. Se ha determinado con claridad que los parámetros relativos a las longitudes de ventana, intervalos de tiempo entre tramas y los períodos de oportunidad para transmitir ofrecen muchas posibilidades de cara a optimizar la funcionalidad de la arquitectura de interfaces. A modo de ejemplo, se podría establecer una configuración dinámica de estos parámetros en función de la información provista por las capas de aplicación y del entorno de operación, para dar mayor prioridad a una clase respecto a otra bajo circunstancias especiales de la comunicación (si requiere tiempo real, si es orientada a conexión, etc.). En la actualidad el borrador del estándar IEEE 802.11p recomienda unos valores predeterminados para estos parámetros de configuración, aunque en el documento oficial se sugiere modificar sus valores para adaptar el funcionamiento al entorno de operación concreto.

A modo de conclusión, el nivel de implementación que se ha alcanzado con la presente arquitectura en VHDL se puede extender bastante más. En cuanto al trabajo futuro, con caracter inmediato se trabaja en la obtención de los retardos asociados a los tiempos de acceso al medio para una versión mejorada de la implementación, que permite configurar el sistema con parámetros más realistas (incluidos los referentes al estándar IEEE 802.11p), y al mismo tiempo permite la integración de la funcionalidad de colas de paquetes, lo que incrementa notablemente el realismo de los resultados. Además se pretende continuar con el desarrollo de un dispositivo completo que implemente la totalidad de los protocolos de la arquitectura WAVE y permita realizar en un futuro mediciones de campo que aporten mejoras cuantitativas respecto de la simulación convencional.

Agradecimientos

Este trabajo ha sido financiado por el proyecto nacional TEC2007-67966-01/TCM (CON-PARTE-1), el proyecto Regional 00002/CS/08 de la Fundación Séneca y está también enmarcado en el Programa de Ayudas a Grupos de Excelencia de la Región de Murcia, Fundación Séneca".

Referencias

[1] Hua Zhu, Ming Li, I. Chlamtac, and B. Prabhakaran. A survey of quality of service in ieee 802.11 networks. *Wireless Communications, IEEE*, 11(4):6–14, Aug. 2004.

[2] IEEE-SA Standards Boards, editor. *IEEE Standard for Information technology, Telecommunications and information exchange between systems, Local and metropolitan area networks, Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Computer Society, 2007.

[3] Frost and Sullivan Consulting. Wireless industry statistics. Disponible en http://www.callcentermagazine.com/article/COM20010822S0007. Acceso en Marzo 2009.

[4] IEEE-SA Standards Boards, editor. *IEEE 1609/802.11p - Family of Standards for Wireless Access in Vehicular Environments (WAVE)*. IEEE Computer Society, 2007.

[5] Xilinx Corporation. Spartan-3e starter kit. Disponible en www.xilinx.com/products/devkits/HW-SPAR3E-SK-US-G.htm. Acceso en Marzo 2009.

[6] EDA Industry Working Groups. Vhdl international (vi). Disponible en http://www.vhdl.org/. Acceso en Abril 2009.

[7] ABIresearch. Dedicated short-range communications (dsrc). Disponible en http://www.abiresearch.com/research/1000690-Dedicated_Short_Range_Communications_(DSRC). Acceso en Marzo 2009.

[8] E. Ferro and F. Potorti. Bluetooth and wi-fi wireless protocols: a survey and a comparison. *Wireless Communications, IEEE*, 12(1):12–26, Feb. 2005.

[9] Tien-Shin Ho and Kwang-Cheng Chen. Performance analysis of ieee 802.11 csma/ca medium access control protocol. volume 2, pages 407–411 vol.2, Oct 1996.

[10] Y. Kim, H. Jung, H. H. Lee, Electron. Cho K. R. Router Technol. Dept., and Taejon; Telecommun. Res. Inst. Mac implementation for ieee 802.11 wireless lan. *Personal Communications, IEEE*, pages 191–195, Apr 2001.

[11] F. Manavi and Y.R Shayan. Implementation of ofdm modem for the physical layer of ieee 802.11a standard based on xilinx virtex-ii fpga. *Personal Communications, IEEE, Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, 3(6):1768–1772, May 2004.

[12] E. Grass, K. Tittelbach-Helmrich, U. Jagdhold, A. Troya, G. Lippert, O. Kruger, J. Lehmann, K. Maharatna, K.F. Dombrowski, N. Fiebig, R. Kraemer, and P. Mahonen. On the single-chip implementation of a hiperlan/2 and ieee 802.11a capable modem. *Personal Communications, IEEE*, 8(6):48–57, Dec 2001.

[13] G. Panic, D. Dietterle, Z. Stamenkovic, and K. Tittelbach-Helmrich. A system-on-chip implementation of the ieee 802.11a mac layer. pages 319–324, Sept. 2003.

[14] Rachida Dssouli, Gregor von Bochmann, and Yair Lahav, editors. *SDL '99 The Next Millennium, 9th International SDL Forum, Montréal, Québec, Canada, 21-25 June, 1999, Proceedings*. Elsevier, 1999.

PROTECCIÓN INTEGRAL DE SISTEMA DE TRAZABILIDAD RFID MEDIANTE FIRMAS AGREGADAS

Guillermo Azuara Guillén, José Luis Salazar Riaño
Departamento de Ingeniería Electrónica y Comunicaciones.
Universidad de Zaragoza.
C/ Maria de Luna 3, 50018 Zaragoza
gazuara@unizar.es, jsalazar@unizar.es

***Resumen*- En el trabajo que se presenta a continuación se detalla cómo puede contribuir la utilización de firmas agregadas a la protección integral de los sistemas RFID. Tras una breve descripción de un sistema de trazabilidad de productos basado en la tecnología RFID y securizado mediante la utilización de firmas agregadas, se hace una revisión de cuáles son las principales amenazas de seguridad para este tipo de sistemas, y se muestra cómo se utilizan las propiedades de dichas firmas para ofrecer una protección integral. A lo largo del trabajo se explica qué medidas de protección hemos adoptado frente a las amenazas descritas en nuestro prototipo real.**

***Palabras Clave*- RFID, Virus, Firmas agregadas.**

I. INTRODUCCIÓN

La tecnología RFID está cada vez más presente en múltiples aspectos de nuestra vida cotidiana y en procesos industriales. Con el avance de la computación ubicua, donde se está generalizando el uso de RFID para el marcado de objetos, se presentan numerosas ventajas, pero también aparecen nuevos retos en el horizonte, como la proliferación de ataques y *malware* específico para entornos RFID. En este trabajo nos centraremos en cómo pueden afectar estas amenazas a un sistema de trazabilidad cárnica basado en RFID y cómo la utilización de firmas agregadas pueden contribuir a prevenir o mitigar sus efectos.

En el siguiente punto se presentará brevemente un sistema de trazabilidad securizado mediante RFID en el que se desea estudiar el impacto de diferentes amenazas.

En el punto tres se presentarán los riesgos de seguridad y las características básicas de amenazas que pueden afectar a los sistemas RFID y sus formas de ejecución.

En el cuarto punto se evaluarán cuáles de estas amenazas pueden afectar al sistema y cómo pueden contribuir las firmas agregadas y otras contramedidas a minimizar su impacto.

En el último apartado se presentarán las conclusiones obtenidas.

II. TRAZABILIDAD RFID

La tecnología RFID permite la identificación de objetos o personas mediante un único identificador, el cuál es transferido con un determinado protocolo hasta un dispositivo receptor, denominado lector, mediante ondas de radio [1]. La forma más simple de RFID consta de una etiqueta (que incluye un minúsculo circuito integrado con su correspondiente antena) y un lector RFID, aunque lo más habitual es conectar este sistema básico a un sistema de información (*back-end*) con una base de datos que almacene toda la información relacionada con el objeto. Es aquí donde esta tecnología puede desplegar todas sus potencialidades, pero también donde se presta a más vulnerabilidades en el ámbito de la seguridad.

La trazabilidad se define, según la Organización Internacional de Estándares (ISO), como "La propiedad del resultado de una medida o del valor de un estándar donde éste pueda estar relacionado con referencias especificadas, usualmente estándares nacionales o internacionales, a través de una cadena continua de comparaciones todas con incertidumbres especificadas" [2].

Cada vez es más habitual encontrar sistemas de trazabilidad basados en la tecnología RFID, que presenta multitud de ventajas frente a otras implementaciones (códigos de barras, por ejemplo). En [3] se propone un sistema de trazabilidad basado en RFID, que permite realizar además de la trazabilidad, un control de requisitos en cada una de las etapas que debe superar el producto, y localizar la entidad responsable de la certificación en cada punto de control. Para ello se recurrió a la utilización de firmas agregadas, que se definen como "una función en la que dado un conjunto de U usuarios, cada uno con su clave pública y privada (Ku_+ y Ku_-), y un subconjunto $V \subseteq U$, si cada usuario $u \in V$ produce una firma σ_u de un mensaje M_u estas firmas pueden ser compactadas en una firma agregada σ por una tercera parte no confiable diferente de los usuarios de V" [4].

La utilización de las firmas agregadas permite verificar con una sola operación criptográfica la validez de los datos grabados en la tarjeta. Esto se realizará en cada punto de control antes de introducir nuevos datos y actualizar la nueva firma agregada. Si todo es correcto se realizarán los cálculos correspondientes y se grabarán los datos correspondientes tanto en la etiqueta RFID como en el sistema de información.

A continuación en la Fig.1, se muestra el esquema del proceso sobre el cual se desea poder realizar la trazabilidad de productos, señalando los hitos o puntos de control que se encargará de garantizar el sistema objeto de nuestro estudio.

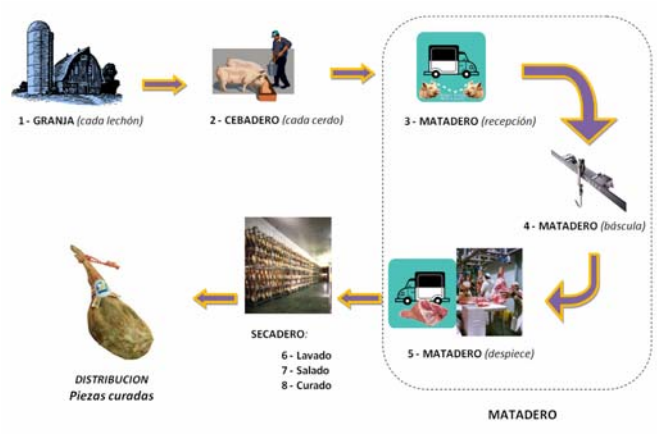


Fig. 1. Esquema del proceso productivo.

En la Fig. 2 se presenta la arquitectura del sistema mencionado, donde cada computador conectado a un lector RFID posee una clave privada, y unos datos de localización geográfica.

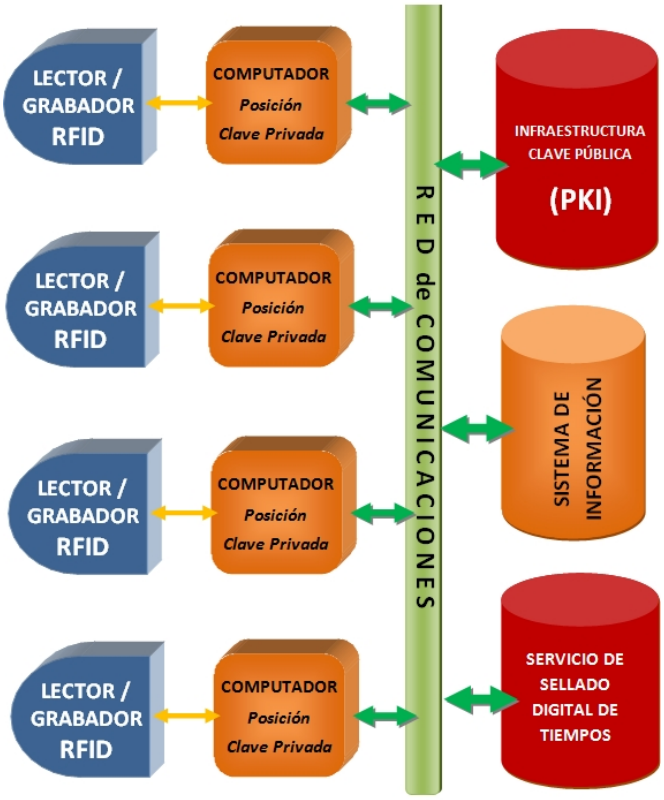


Fig. 2. Arquitectura del sistema de trazabilidad.

En cada punto de control el lector tomará los datos de la tarjeta que se pasarán al computador. Éste comprobará la firma agregada y si es correcta generará una nueva firma que englobe los datos insertados en ese punto, pasará los nuevos datos y la nueva firma al sistema de información. Esta operación será ejecutada por lo que denominamos "módulo de seguridad" que es la parte del sistema encargada de generar y verificar las firmas agregadas en cada punto. Por tanto, este sistema actuará como filtro de los datos, evitando que los que no estén firmados correctamente alcancen el sistema de información.

III. Riesgos de seguridad en RFID

Son muy variados los tipos de ataques que pueden sufrir los sistemas RFID. Esta tecnología no está exenta de riesgos, y algunos autores han presentado las amenazas más importantes agrupadas bajo diversas taxonomías. Muchas de ellas centran esta clasificación basada en las amenazas relacionadas con la privacidad ([5], [6] y [7]). En nuestro sistema, la privacidad no es algo que nos interese salvaguardar en el proceso, ya que de hecho, la trazabilidad en este caso es el objetivo de todo el sistema. Por otro lado, en [8] se propone una taxonomía que añade tres grupos más de riesgos además del ya mencionado de la privacidad: riesgos de "procesos de negocios" (*Business Process Risks*), relacionados con el impacto de fallos del sistema de RFID en sistemas automáticos basados en él; riesgos de "inteligencia de negocios" (*Business Intelligence Risks)* y finalmente riesgos "externos". Aunque esta perspectiva es más extensa y cercana a la realidad que las anteriores ya que abarca más tipos de amenazas, nos vamos a basar en la estructura propuesta en [9], en donde se clasifican los tipos de ataques y amenazas relacionándolos con un modelo de cuatro capas: física, red y transporte, aplicación y estratégica. Las tres primeras se corresponden bastante bien con las capas correspondientes del modelo OSI, la cuarta engloba los riesgos relacionados con factores logísticos y además también se plantea la posibilidad de que los ataques sean multi-capa, es decir, afecten a varias de las capas básicas.

Como se desarrolla en [9], los principales riesgos de cada capa son:

- Capa física: inhibición permanente de las etiquetas, inhibición temporal de las tarjetas y ataques de repetición.
- Capa de red y transporte: suplantación de la identidad (afectarían a las tarjetas), suplantación y escuchas no autorizadas (ambas estarían más relacionadas con los lectores), y ataques cuyo objetivo serían los protocolos de red.
- Capa de aplicación: lecturas no autorizadas de las etiquetas, modificación de las etiquetas, ataques relacionados con el *middleware* (como desbordamiento de *buffers* e inyección maliciosa de código).
- Capa estratégica: espionaje industrial, técnicas de ingeniería social, amenazas a la privacidad, selección de objetivos (por ejemplo detectar personas con artículos de lujo para atracarlos).

Además, también plantean amenazas multicapa, entre las que caben señalar: ataques de denegación de servicio, lectura / escritura de información en el espacio libre de la tarjeta sin conocimiento del usuario, análisis de tráfico, ataques a los algoritmos de cifrado de la información, ataques basados en la monitorización de parámetros físicos de funcionamiento (consumo de energía, variación en los campos electromagnéticos, etc…) y ataques de repetición (grabar una contraseña de una transacción anterior y repetirla cuando se presente el mismo desafío).

IV. ROBUSTEZ DEL SISTEMA FRENTE A LAS AMENAZAS

A la vista de las amenazas planteadas, vamos a estudiar las que probablemente más puedan afectar al sistema, y qué medidas de protección se han adoptado, algunas de ellas basadas en las reglas propuestas en [10].

A. La capa física

Es muy posible que durante el proceso alguna etiqueta se desprenda o se dañe. En este caso, en el primer punto de control en que se detecte el fallo o ausencia de etiqueta se sacará el producto de la cadena de producción, se comprobará hasta donde se ha realizado el proceso en la base de datos, se generará una nueva etiqueta con toda la información y se continuará el proceso desde ese punto.

El comando KILL [11], creado para defender la privacidad del futuro comprador una vez que adquiera el producto y que inutiliza permanentemente la tarjeta, no tiene sentido en este proceso ya que la etiqueta sólo tiene un uso interno y no está previsto que llegue al usuario final. Idéntico tratamiento se sigue para el comando LOCK [6], creado para una inhabilitación temporal o permanente de la escritura.

Respecto a las interferencias activas, parece complicado que un supuesto atacante las pudiera crear, ya que el proceso se realiza siempre en instalaciones privadas, por lo que será suficiente con observar las medidas de protección físicas propuestas en [12] como son: el control de accesos, cámaras de seguridad, vigilantes y medidas en esta línea. No obstante, se deberá disponer de un plan de contingencia para en el caso de que algún punto de control del sistema de trazabilidad falle, se pueda seguir realizando el proceso productivo y de toma de datos.

La posibilidad de ataques de repetición, como el propuesto en [13], es complicado debido a la propia naturaleza de las instalaciones y a las cortas distancias de comunicación entre etiquetas y lectores. Por otro lado, el uso de firmas agregadas, protege al sistema de una posible alteración maliciosa de los datos, ya que todos van protegidos por la firma y un sellado temporal. Por todo ello es altamente improbable una alteración de los datos sin que el sistema la detecte.

Al hilo de lo anterior, la amenaza más preocupante por su impacto sería la de una clonación de etiquetas. Aunque clonarla con la misma apariencia física es complicado, ya que implicaría fabricar idéntico modelo de tarjeta con el mismo número de identificación, y por definición es algo que no van a hacer los fabricantes. Sí que es cierto que existen etiquetas reprogramables, pero si son distintas estéticamente a las legítimas no sería factible su utilización. Además, estas etiquetas clonadas deberían introducirse en el mercado cuando el producto acabe el proceso de producción, ya que de lo contrario sería fácilmente detectable por el sistema. Aún así y si se desea aumentar el nivel de seguridad, se podría recurrir a soluciones hardware basadas en la utilización de funciones físicamente inclonables (PUFs) ([14] y [15]). En el caso de que se clonaran etiquetas similares pero con distinto número de identificación único serían inmediatamente detectadas ya que la firma agregada incluye el identificador único, y por tanto si la tarjeta no es la misma el identificador cambia y es automáticamente detectado que la firma no es correcta.

B. La capa de red y transporte

En la etiqueta no hay ningún dato cifrado, por lo que no habría ningún problema en que alguien leyera el contenido de la tarjeta haciéndose pasar por un lector legítimo. En el mismo sentido, si un lector fraudulento escribiera una tarjeta, sería detectado en el siguiente punto de control, dado que no sería capaz de realizar una firma válida.

C. La capa de aplicación

Como se ha comentado en el punto anterior, no existe ningún problema con las lecturas no autorizadas ya que toda la información va en texto plano, y como también se ha señalado, la modificación de los datos es automáticamente detectada por el módulo de seguridad.

Aunque en la clasificación de riesgos que estamos utilizando se habla exclusivamente de desbordamientos de *buffer* e inyección maliciosa de código, podríamos hablar en general de *malware* específicamente diseñado para atacar los sistemas RFID [16], es decir software diseñado para causar daños en los sistemas. Dentro del *malware* tenemos tres tipos principales: explotación de vulnerabilidades conocidas de RFID (*RFID exploits)*, gusanos RFID y Virus RFID [16].

Partiendo de las principales amenazas citadas por [16], el mayor peligro de los *exploits* en RFID, es precisamente que muchas veces no son esperados, y su procesamiento puede explotar vulnerabilidades tanto del sistema RFID, como del propio sistema de información o la red entera. Normalmente los objetivos de estos ataques son componentes específicos del sistema, como las bases de datos, interfaces web o las APIs de gestión de los lectores. Uno de los ataques más sencillos sería la inyección de comandos SQL como *shutdown* o *drop table* que causarían graves perjuicios al sistema (apagado y eliminación de tablas) o incluso el robo de datos. Por ejemplo, algunas bases de datos, como Microsoft SQL Server permiten ejecutar comandos de sistema a los administradores, por lo que si la base de datos está funcionando como *root* se podría comprometer todo el sistema. El uso de las firmas agregadas garantiza que los datos han sido introducidos por entidades autorizadas, por lo que si la firma no es correcta no se pasarán los datos al sistema de información, y el módulo de seguridad descartará los datos y por tanto no ejecutará el comando.

Otro tipo de ataque dentro de los *exploits* englobaría la inserción de código. Si las aplicaciones RFID utilizan protocolos web para hacer consultas a la base de datos, es muy probable que los clientes sean capaces de interpretar *scripts*, en cuyo caso podrían ser vulnerables a este tipo de ataques al igual que los navegadores web, pero de mayor peligrosidad, ya que normalmente los navegadores tienen limitado el acceso al *host.* Para prevenir esto, y dado que normalmente los códigos citados utilizan caracteres distintos a número y letras, una primera medida sería que el sistema no procesara ninguna entrada que poseyera algún elemento distinto a los citados. Adicionalmente, como en el caso anterior, el módulo de seguridad filtrará los datos que llegan al sistema de información. Si es posible sería bueno no permitir la ejecución de lenguajes *script* en el sistema *back-end*.

No menos peligroso es un desbordamiento de *buffer* (*buffer overflow*) que se produce cuando en un área de memoria se escriben más datos de los que puede contener, y estos sobrescriben zonas de memoria anexa. El mayor

peligro de esto es que un atacante produciendo intencionadamente un desbordamiento, puede ejecutar código arbitrario. Aunque la mayoría de los compiladores, navegadores y sistemas operativos ya están protegidos contra estos ataques, en [16] se presenta un ejemplo de desbordamiento en un sistema RFID. En este caso, el filtrado previo del módulo de seguridad a través de la confirmación de la validez de la firma, hace que los datos maliciosos no pasen al sistema. También se debe ser especialmente cuidadoso en la programación, y asegurarse de leer sólo las zonas de memoria donde se hallan los datos.

Los gusanos RFID se basan en la explotación de fallos de los sistemas para introducir código malicioso en el lector que sobrescribirá las tarjetas con un código que cuando sean leídas provocará la infección de un nuevo lector que a su vez infectará nuevas tarjetas. Nuevamente la comprobación previa de las firmas impedirá la propagación de la infección mediante las tarjetas, pero se deberá auditar la seguridad del sistema para saber cómo ha sido infectado el lector (probablemente vía red). En este caso, es importante volver a señalar la importancia de una política de seguridad adecuada que afecte a todos los sistemas implicados en el proceso.

La principal diferencia entre un gusano y un virus es que el gusano no necesita la intervención de un usuario para propagarse [17]. El primer virus RFID fue presentado y explicado profusamente en [18], y realiza una inyección de código SQL cuando la etiqueta es leída, copiando las instrucciones de inyección en la base de datos, de manera que cuando una nueva etiqueta sea escrita quedará infectada y también propagará el virus. A partir de esta idea inicial, se pueden programar *payloads* para causar efectos más peligrosos. Como mejoras al virus descrito, y por tanto con un grado mayor de amenaza y peligrosidad, la autora de [18] plantea la posibilidad de dar polimorfismo a los virus, mejorar sus técnicas de ocultación y dotarlos de la mayor generalidad posible.

Incluso tratándose de virus con las mejoras descritas, aplicando la normas generales básicas habituales como limitar los permisos de la base de datos y de los usuarios, aislar el servidor *middleware* del resto de la red y revisar el código del *middleware* para evitar agujeros de seguridad [19], contamos con un grado de protección elevado frente a estas amenazas (al menos de momento). Además el "módulo de seguridad" implementado permite limitar totalmente la propagación del virus mediante las tarjetas, ya que los datos no serán procesados por el sistema de información.

D. La capa estratégica.

De los riesgos comentados (espionaje industrial, técnicas de ingeniería social, amenazas a la privacidad, selección de objetivos) sólo nos afectaría, en alguna medida, la posibilidad de que mediante técnicas de ingeniería social se consiguiera que alguna persona con algún privilegio, realizase alguna acción que pudiera comprometer la seguridad del sistema. Las contramedidas propuestas son: la formación adecuada del personal y la definición de una política general de seguridad, así como la imposibilidad de acceso directo a las claves privadas de los dispositivos por parte de los usuarios, es decir, cuando un usuario legítimo inicia la sesión se debe autenticar frente al sistema, luego será el sistema el que realice los cálculos, no teniendo acceso directo el usuario a la clave privada de firma utilizada en la operación.

E. Ataques multicapa.

Los ataques que estimamos pueden afectar a nuestro sistema son: denegación de servicio, la lectura / escritura de información en el espacio libre de la tarjeta sin conocimiento del usuario, ataques de repetición. El resto dado que no es necesaria privacidad no nos afectan.

Los ataques de denegación de servicio podrían afectar a la cadena de producción, no obstante para su ejecución haría falta acceso físico a los recintos, por lo que las medidas físicas propuestas en el subapartado A deberían ser suficientes. No hay que olvidar que dada la existencia de redes de comunicaciones en el sistema, éstas sí serían vulnerables a ataques externos de DoS, por lo que deberán tomarse las medidas de seguridad habituales para evitar este tipo de ataques.

La lectura no autorizada, como ya se ha comentado, no supone ningún problema, ya que no hay datos secretos. La escritura no autorizada ya ha sido tratada tanto en el apartado B como en el C.

En cuanto a los ataques de repetición también ya han sido tratados, concretamente en el subapartado A.

V. CONCLUSIONES

Como se ha expuesto en los puntos anteriores, la seguridad de un sistema RFID depende de la seguridad en todas las capas que integran el sistema, no debiendo descuidar ninguna de ellas.

Dado que en nuestro caso, el sistema siempre va a estar en instalaciones privadas sin acceso público, muchos de los problemas no nos van a afectar. Además, dado que no necesitamos privacidad los ataques que buscan obtener datos secretos tampoco deben preocuparnos.

El uso de las firmas agregadas en el sistema, además de la función inicial de asegurar la veracidad de los datos, la hora de introducción y que sólo entidades autorizadas realicen la escritura, también protege frente a otro tipo de amenazas, como se ha señalado en los puntos anteriores: ataques de repetición, clonación, escritura no autorizada, modificación de datos de etiquetas, inyección SQL, inserción de código, desbordamiento de *buffer*, gusanos RFID, virus RFID, ingeniería social.

La peor amenaza que se cierne sobre la seguridad RFID es pensar que el sistema es seguro por el mero hecho de utilizar tecnología RFID, siendo fundamental el conocimiento de los riesgos existentes y la implementación de todas las contramedidas que sea posible, tanto en la parte de RFID como en todo el resto de sistemas utilizados en el proceso.

AGRADECIMIENTOS

Este trabajo ha sido apoyado por el Instituto Nacional de Investigación y Tecnología Agraria y Agroalimentaria (INIA) a través del proyecto PET2007-08-C11-06.

REFERENCIAS

[1] "The History of RFID Technology", *RFID Journal* [Online]. Disponible en: http://www.rfidjournal.com/article/articleview/1338/1/129/. Última consulta: Marzo 2009.

[2] "ISO/IEC Guide 99:2007", *International Organization for Standarization (ISO)*, 2007.

[3] G. Azuara, J.J. Piles, J. L. Salazar, "Securización de un sistema de trazabilidad RFID mediante firmas agregadas", *VII Jornadas de Ingeniería Telemática,* Libro de Actas, ISBN: 978-84-612-5474-3, pp-57-63, Alcalá de Henares, Septiembre 2008.

[4] D. Boneh, C. Gentry, B. Lynn, H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps", en Cryptology ePrint Archive, Report 2002/175, Volume 2656 de *Lecture Notes on Computer Science*, (2002).

[5] Avoine, G., Oechslin, P., "RFID Traceability: A Multilayer Problem". En Patrick, A., Yung,M. (eds.). *Proc. of the Ninth Int'l Conf. on Financial Cryptography and Data Security (FC'05)*, Lecture Notes in Computer Science, Vol. 3570. Pp. 125-140, (2005).

[6] Ayoade, J., Saxby, S., "Roadmap for Solving Security and Privacy Concerns in RFID Systems.", *Computer Law and Security Report* , (2007)

[7] Garfinkel, S., Juels, A., Pappu, R., "RFID Privacy: An Overview of Problems and Proposed Solutions", *IEEE Security & Privacy*, Vol. 3, pp. 34-43, (2005).

[8] Karygiannis, A., Phillips, T., Tsibertzopoulos, A., "RFID Security: A Taxonomy of Risk.", *Proceedings China'Com '06*, pp. 1.8, (2006).

[9] A. Mitrokotsa, M.R. Rieback, A.S. Tanenbaum, "Classication of RFID Attacks", *2nd Intl. Workshop on RFID Technology (IWRT'08)*, pp. 73-86, Barcelona, Spain, Junio 2008.

[10] D. Rajesh., "Advanced concepts to prevent SQL injection". http://www.csharpcorner.com/UploadFile/ajeshdg/Page1071420050529 57AM/P%age1.aspx?ArticleID=631d8221-64ed-4db7-b81b-8ba3082cb 496. [on-line].

[11] Center, A.I., "900 MHz Class 0 Radio Frequency (RF) Identification Tag Specification". Draft, www.epcglobalinc.org/ standards/specs/900 MHz Class 0 RFIDTag, (2003)

[12] Karygiannis, T., Eydt, B., Barber, G., Bunn, L., Phillips, T., "Guidelines for Securing Radio Frequency Identification (RFID) Systems", *NIST Special Publication 800-98*, National Institute of Standards and Tecnology, (2007).

[13] Kfir, Z., Wool, A., "Picking Virtual Pockets Using Relay attacks on Contactless Smartcard", *Proc. of the 1st Int'l Conf. on Security and Privacy*, pp. 47-48, (2005).

[14] Bolotnyy, L., Robins, G., "Physically Unclonable Function-Based Security and Privacy in RFID Systems", *Proceedings of PerCom'07*, pp. 211-220, New York, USA, (2007).

[15] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, V. Khandelwal, "Design and implmeeentation of PUF-Based "Unclonable" RFID ICs for Anti-Counterfeiting and Security Applications", *2008 IEEE International Conference on RFID*,Las Vegas, USA, April 2008.

[16] Rieback, M., "Security and privacy of radio frecuency identification", *Thesis Vrije Universiteit*, Amsterdam, 2008.

[17] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham, "A taxonomy of computer worms", *First Workshop on Rapid Malcode (WORM)*, 2003.

[18] M. Rieback, B. Crispo, and A. Tanenbaum, "Is your cat infected witha computer virus?", *IEEE Pervasive Computing and Communications*, pp. 169-179, Pisa, Italy, March 2006.

[19] D. Rajesh, "Advanced concepts to prevent SQL injection", http://www.csharpcorner.com/UploadFile/rajeshdg/Page107142005052 957AM/P%age1.aspx?ArticleID=631d8221-64ed-4db7-b81b-8ba3082c b496.

IMPACTO DE MECANISMOS DE SEGURIDAD EN SENSORES IEEE 802.15.4

Carolina Tripp, Jordi Casademont Serra
Departamento de Ingeniería Telemática
Universidad Politécnica de Cataluña (UPC)
C/Jordi Girona 1-3. 08034 Barcelona, España
ctripp@entel.upc.edu

Resumen **- En la actualidad son muchos los mecanismos de seguridad que el estándar IEEE 802.15.4 permite a las redes inalámbricas de sensores [1] Dicho estándar define las especificaciones de la Capa de Acceso al Medio y la Capa Física de los dispositivos inalámbricos de área personal. La última revisión corresponde al 2006. Dichas revisiones y actualizaciones son hechas por el grupo de trabajo 802.15.**

Sin embargo estos mecanismos consumen recursos como memoria y batería, que son un poco limitados en estos dispositivos. Además de contribuir a los retardos en la comunicación. Por lo cual, en el presente trabajo se presenta de manera práctica el impacto que el uso de mecanismos de seguridad tienen en el desempeño de este tipo de redes. Para ello se hizo una comparación de dicho desempeño de manera teórica basándose en lo presentado en [2], con los valores óptimos apegados a lo especificado en el estándar IEEE 802.15.4, en contraste con pruebas reales. Para estas pruebas se hizo uso del sistema operativo TinyOS [3] y de las operaciones de seguridad MAC (Capa de Acceso al Medio) ofrecidas por el chip CC2420 usado en las motas TelosB. Además se presenta el desgaste de la batería, el cual es otro punto importante que se desea conservar en los sensores.

Palabras clave **- Sensores, TinyOS, MAC, TelosB.**

I. INTRODUCCION

Desde hace varios años ya, las comunicaciones inalámbricas pasaron a ser parte de la vida cotidiana, hasta el punto de que en la actualidad estamos completamente conectados, ya sea con nuestro móvil o cualquier otro dispositivo inalámbrico con el cual estamos en constante envío o recepción de información.

Los avances actuales en las comunicaciones inalámbricas son las que han permitido que estos dispositivos puedan desarrollarse. Gracias al estándar IEEE 802.15.4 [4] es posible su conectividad, ya que define las características de la Capa Física (PHY) y la Capa de Acceso al Medio (MAC) para los dispositivos de área personal (LR-WPAN, *Low-Rate Wireless Personal Area Networks*).

Sin embargo, este tipo de dispositivos son susceptibles, al igual que las redes inalámbricas tradicionales al ataque sobre la información transmitida. Es por ello que asegurar la información es una de las principales preocupaciones, ya que el canal de comunicación no requiere la participación física de un cable. Y debido a sus características (baja potencia, reducida capacidad de procesamiento y memoria) hacen muy difícil el uso de métodos criptográficos conocidos.

El SmartRF CC2420 [5] es un chip IEEE 802.15.4/ZigBee, operando en la banda de 2.4 GHz con velocidad de datos de 250 Kbps. Es actualmente uno de los chips más populares para trabajar en redes inalámbricas de sensores. CC2420, tiene el soporte de hardware para el formato de trama del estándar IEEE 802.15.4.

Una de las principales características del chip CC2420 es el de soportar operaciones de seguridad, como cifrado, descifrado, autenticación e integridad. Es capaz de realizar dichas operaciones a nivel MAC, entre las cuales se incluyen CTR (*Counter Mode)* que proporciona cifrado, CBC-MAC (*Cipher Block Chaining-Message Authentication Code)* el cual permite comprobar la autenticidad del emisor y la integridad del mensaje y CCM (*Counter Mode with CBC-MAC*) el cual es una combinación de los dos anteriores. Cada unas de estas, basadas en el cifrado AES [6] (*Advanced Encryption Standard*) usando claves de 128 bits.

El presente trabajo está organizado en cuatro apartados, los cuales presentan una breve descripción de la seguridad que puede usada en las motas de las pruebas, seguido por la la implementación de la misma. Continuando con la presentación de los resultados, para finalizar con la presentación de los resultados finales de las pruebas.

II. SEGURIDAD IN-LINE

CC2420 puede realizar operaciones de seguridad (cifrado, descifrado, autenticación e integridad) a nivel MAC dentro de las tramas TxFIFO (transmisión) y RxFIFO (recepción). Estas operaciones son llamadas operaciones de seguridad *In-line* [5].

Los distintos modos de trabajar son:

- ✧ Sin seguridad
- ✧ CTR (cifrado / descifrado)
- ✧ CBC – MAC (autenticación e integridad)
- ✧ CCM (autenticación, integridad y cifrado / descifrado)

III. IMPLEMENTACION

Para la realización de las pruebas prácticas, se elaboró un pequeño programa que realizaba envíos de datos de un sensor a otro, uno programado como estación base o receptor y otro como simple emisor, tomando como base un ambiente ideal, es decir solo una fuente y un receptor, por lo cual no había colisiones. Se consideró el tamaño máximo de datos de usuario en cada caso. Puesto que no se debe superar el *payload* permitido (127 bytes) ya que los sensores no realizarían ninguna operación.

Se usó la implementación de seguridad CC2420 soportada para TinyOS 2, la cual permitió la evaluación de la red con respecto a su desempeño bajo CTR (cifrado / descifrado), CBC – MAC (autenticación e integridad) y CCM (integridad, autenticación y cifrado / descifrado). En el caso de CBC y CCM pudiendo usarse distintos tamaños para el código MAC (4, 8 ó 16).

Esta implementación permite elegir entre los tres modos de seguridad, en TinyOS la función fue llamada SecAMSend y provee una interface que proporciona estas opciones, mediante los comandos:

- ✧ call CC2420Security.setCtr(*key*, *payload*);
- ✧ call CC2420Security.setCbcMac(*key*, *payload*, *MAClen*);
- ✧ call CC2420Security.setCcm(*key*, *payload*, *MAClen*);

Ellas habilitan la seguridad seleccionada antes del envío de cada paquete.

El primero ***key,*** permite al usuario seleccionar la clave, recordando que hay espacio en memoria para dos claves.

El segundo parámetro ***payload***, establece el número de bytes del *payload* que se desea no se tomen en cuenta para el cifrado (en el caso de CTR y CCM) y la autenticación (en el caso de CBC y CCM). Por defecto este parámetro es 0, ya que lo normal es iniciar luego de las cabeceras y tomar en cuenta todo el *payload* para iniciar estas operaciones.

El tercer parámetro ***MAClen***, es usado en CBC-MAC y CCM para especificar la longitud del código MAC (código de autenticación de mensaje) [7]. Los valores pueden ser elegidos entre los valores 4, 8 y 16.

Una vez indicados los valores respectivos en las funciones, se puede llamar la interfaz de envío AMSend de manera normal.

IV. RESULTADOS

Primero se comprobó que las cabeceras añadidas dependiendo del mecanismo usado fueron los correctos. En la Tabla 1 se puede ver la cantidad del *payload* ***n*** (datos enviados) y la longitud final de la trama según el mecanismos implementado.

Con los datos de la Tabla 1 podemos ver cuál es la longitud máxima (en bytes) de datos de usuario que pueden ser enviados dependiendo del tipo de seguridad que se desea implementar, pues se debe recordar que el tamaño máximo de la trama (incluyendo cabeceras) no podrá ser mayor a 127 bytes. En la tabla aparecen en negritas aquellas tramas que sobrepasan esta regla. De esta manera podemos darnos cuenta que mientras Sin Seguridad pueden enviarse 113 bytes de datos de usuario, en el caso de CCM16 el máximo sería de tan solo 92 bytes.

n	**NO SEC**	**CTR**	**CBC-MAC-4**	**CBC-MAC-8**	**CBC-MAC-16**	**CCM4**	**CCM8**	**CCM16**
60	74	79	78	82	90	83	87	95
65	79	84	83	87	95	88	92	100
70	84	89	88	92	100	93	97	105
75	89	94	93	97	105	98	102	110
80	94	99	98	102	110	103	107	115
85	99	104	103	107	115	108	112	120
90	104	109	108	112	120	113	117	125
95	109	114	113	117	125	118	122	**130**
100	114	119	118	122	**130**	123	127	**135**
105	119	124	123	127	**135**	**128**	**132**	**140**
110	124	**129**	**128**	**132**	**140**	**133**	**137**	**145**
115	**129**	**134**	**133**	**137**	**145**	**138**	**142**	**150**

Tabla 1. Longitud final de tramas.

A. *Impacto de la Seguridad en los Retardos de Transmisión*

Se hizo la programación de las motas TelosB [8], estas hacían el envío de mil tramas con diferentes longitudes de datos de usuario, desde 60 bytes hasta el máximo soportado por cada mecanismo de seguridad. Esto nos daba el tiempo total del envío de estas tramas y a partir de esos resultados se pudo promediar el tiempo en milisegundos para cada una de ellas.

En los resultados mostrados en la Tabla 2 se puede ver que la diferencia de tiempos entre enviar una trama sin cifrar y una trama cifrada (CTR) es 0.72 milisegundos, con respecto a una autenticada (CBC MAC-16) es de 1.5 milisegundos, y con respecto a una trama cifrada y autenticada (CCM16), la diferencia incrementa a 1,99 milisegundos.

Como se puede observar en las Figuras 1 y 2, la distribución en cuanto a resultados es bastante parecida, más no los rangos obtenidos. Es decir que se puede observar claramente la diferencia entre una trama sin ningún tipo de seguridad, la cual no tiene *overhead* extra ni procesos añadidos al envío en comparación con aquella a la cual se le ha implementado algún tipo de seguridad.

Pero en cuanto a los rangos de throughput se puede observar una diferencia de alrededor del 7 y 16% entre el caso teórico y el caso práctico. Esto está dado por los tiempos de *backoff*. Es decir, en el caso teórico se considera una *backoff* fijo independiente de los envíos, de los tamaños de trama y de cualquier otro factor que se presentara. Esto no fue así al realizar las pruebas reales. Ya que TinyOS está programado para que los *backoff* sean tan aleatorios como sean posibles y esto hace que los *backoff* no sean fijos. Esto provocó que el throughput tenga este decremento considerable.

Bytes	No-SEC	CTR	CBC-MAC-4	CBC-MAC-8	CBC-MAC-16	CCM-4	CCM-8	CCM-16
60	9,31	10,03	9,85	10,17	10,81	10,36	10,67	11,30
65	10,00	10,46	10,36	10,67	11,29	10,81	11,17	11,79
70	10,45	10,81	10,74	11,08	11,77	11,17	11,47	12,18
75	10,80	11,23	11,14	11,48	12,19	11,57	11,98	12,59
80	11,21	11,67	11,60	11,52	12,60	12,03	12,32	13,00
85	11,64	12,14	11,95	12,32	13,01	12,46	12,82	13,44
90	12,14	12,63	12,45	12,80	13,42	12,97	13,31	13,97

Tabla 2. Retardos en los envíos en mseg.

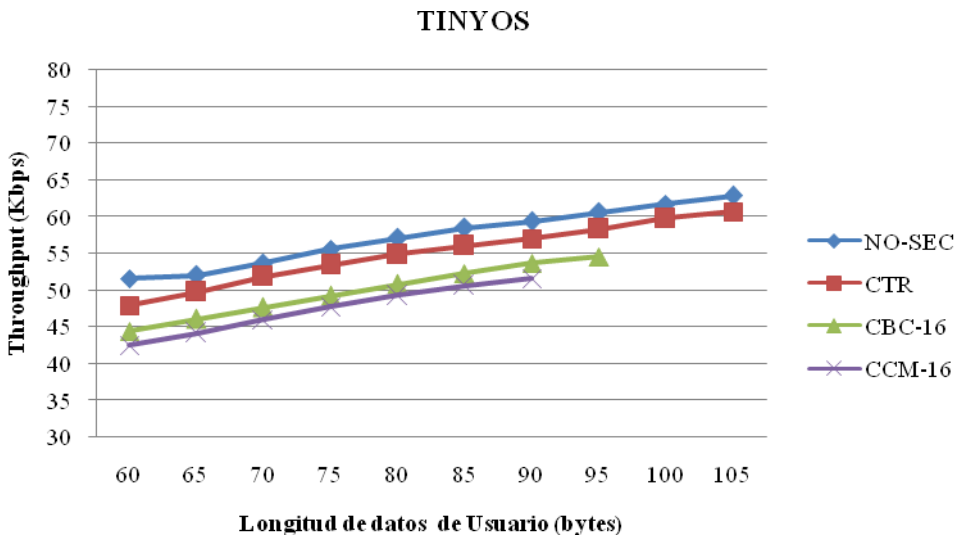


Fig. 1. Throughput efectivo, caso práctico (TinyOS). NO ACK.

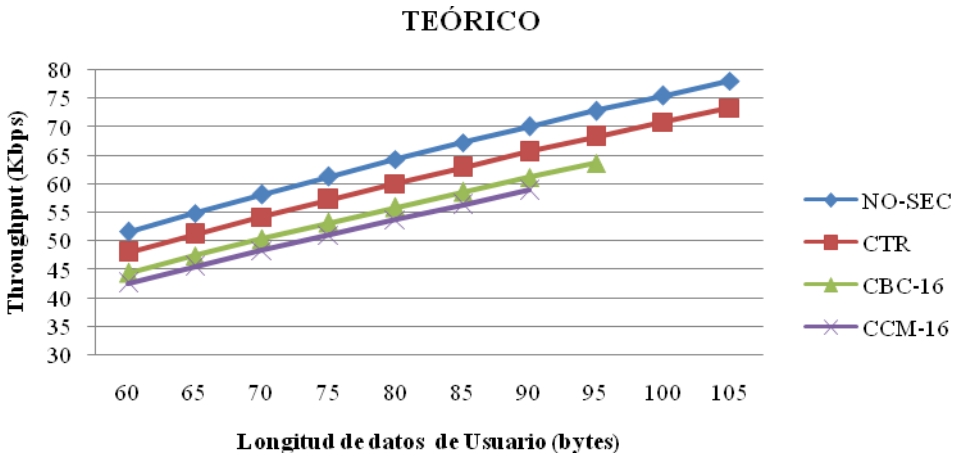


Fig. 2. Throughput efectivo, caso teórico. NO ACK.

B. Impacto de la Seguridad en el Consumo de Energía

En el caso de la energía requerida, se hizo uso de un analizador de potencia para poder observar el comportamiento de consumo de la carga al momento que los sensores están transmitiendo o recibiendo información. Esto haciendo diferentes lecturas, se inició haciendo pruebas cuando los sensores trabajan sin ningún tipo de seguridad, en decir solo enviando datos a una estación base.

Este analizador funciona como una fuente de alimentación de energía, que permite la medición de voltaje e intensidad. Las mediciones se hicieron tanto en el nodo transmisor como aquel programado como Estación Base, el cual recibe toda la información que el emisor está generando.

Se realizó una prueba con las motas programados para envíos de tramas de 90 bytes de usuario sin seguridad, pasando a la misma cantidad de envíos con la misma longitud de datos de usuario pero bajo la implementación de CCM16 y finalizando con un tiempo en reposo, es decir, sin actividad alguna.

Como resultado de las lecturas con el analizador se pudo obtener que como media, la corriente usada para transmitir paquetes sin cifrar es de 24 mA y en recepción 26 mA. En el caso del envío de datos bajo la seguridad CCM16 la media en los envíos es de 22 mA y de igual manera que en el caso anterior 26mA en recepción. Esto sin hacer uso de ACK y aplicando un voltaje de 3 V. En la Figura 3, se puede distinguir claramente como en los envíos sin seguridad hay una carga de 24 mA, cuando pasa a CCM16 hay una disminución de 2 mA y en el caso de reposo baja hasta 4.7 mA, en el caso de transmisión.

Estos resultados se deben a que en el caso de no presentar el añadido de ningún método de cifrado o seguridad, la carga es constante y el envío se hace con mayor rapidez, requiriendo una energía menor que en el caso contrario.

En el caso de recepción, mientras no recibe ningún paquete se encuentra en un nivel de carga de 24 mA en promedio. Esto porque siempre está en estado de espera, es decir en un constante monitoreo de recibir información, por ello su descenso no es mayor. Esto se puede observar claramente en la Figura 4, en la cual se observa constante la recepción de datos en 26 mA y el pequeño descenso en caso de no recibir nada.

Esto demostró que aunque la carta necesaria en mayor en caso de no usar seguridad, como los envios se hacen en un tiempo menor la energía consumida es menor que en el caso de usar seguridad CCM16.

En la Tabla 3 se resumen el desgaste de energía, el cual depende de los tiempos de envío que se necesita para cada caso, esto se obtuvo anteriormente con el analizador. Se pudo comprobar cómo la recepción con seguridad CCM16 gasta más energía, esto se debe a que en caso de implementar un tipo de seguridad los tiempos de envíos y recepción son mayores.

Una batería convencional usada en las motas, tiene una capacidad de 2000mA·hora, y tomando en cuenta los datos obtenidos anteriormente en cuando a la transmisión de tramas y cargas correspondientes, se puede llegar a la conclusión de que la cantidad de envíos dependiendo la longitud de datos de usuario, lo cual se muestra en la Tabla 4.

Esto es multiplicando el tiempo de transmisión por la carga necesaria, en este caso 22 o 24 mA según corresponda. El valor obtenido esta dado en mA·seg, entonces a continuación se pasa a mA·Hora y se divide entre los 2000 que soporta la batería.

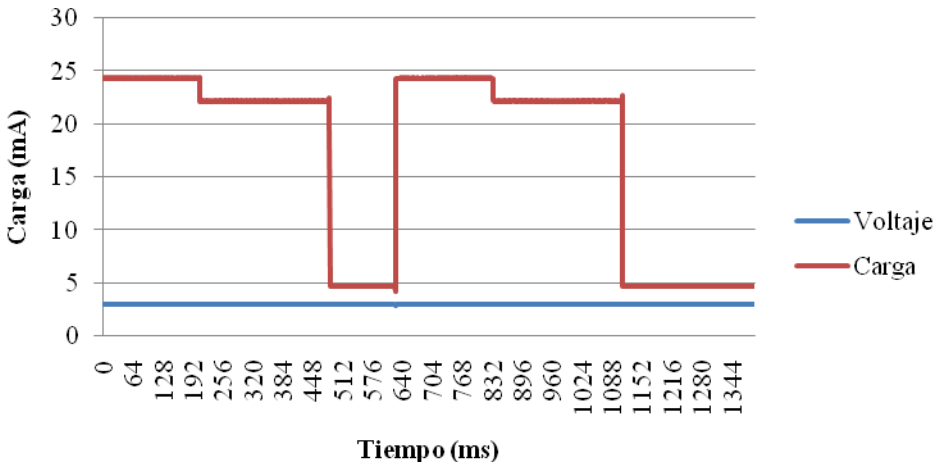


Fig. 3. Transmisión de paquetes Sin Seguridad y CCM16.

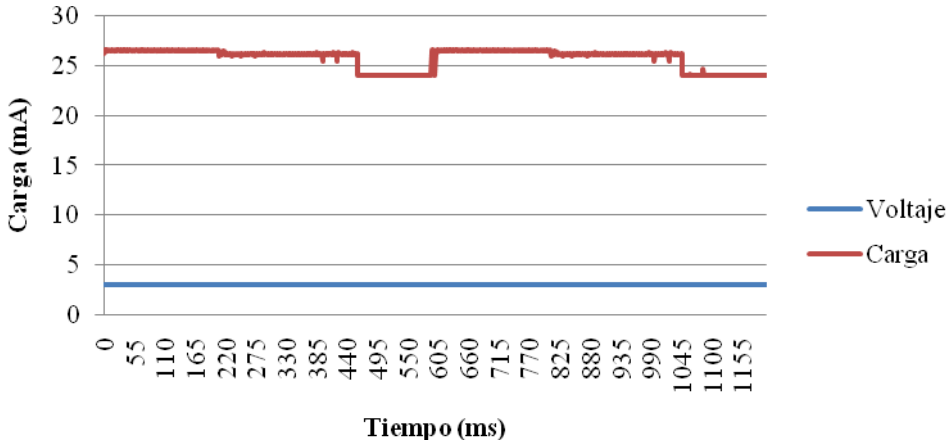


Fig. 4. Recepción de paquetes Sin Seguridad y CCM16.

	NOSEC			CCM16		
bytes	**Tiempo (t)**	**Energía (mJ) TX**	**Energía (mJ) RX**	**Tiempo (t)**	**Energía (mJ) TX**	**Energía (mJ) RX**
60	0,0093	0,6696	0,7254	0,0113	0,7458	0,8814
65	0,0100	0,72	0,78	0,0118	0,7788	0,9204
70	0,0104	0,7488	0,8112	0,0122	0,8052	0,9516
75	0,0108	0,7776	0,8424	0,0126	0,8316	0,9828
80	0,0112	0,8064	0,8736	0,0130	0,858	1,014
85	0,0116	0,8352	0,9048	0,0134	0,8844	1,0452
90	0,0121	0,8712	0,9438	0,0140	0,924	1,092
95	0,0126	0,9072	0,9828	--	--	--
100	0,0130	0,936	1,014	--	--	--
105	0,0134	0,9648	1,0452	--	--	--

Tabla 3. Energía requería en transmisión y recepción.

	Tramas (x 10^6)	
bytes	**NOSEC**	**CCM16**
60	116	98
65	109	93
70	103	89
75	97	85
80	92	81
85	87	78
90	83	75

Tabla 4. Cantidad de tramas transmitidas con una batería AA.

V. CONCLUSIONES

Luego del análisis de los resultados de las pruebas se vio la importancia de tomar en cuentas todas las restricciones en este tipo de redes de comunicación al momento de proporcionar seguridad en ellas, puesto que sus características son muy diferentes a las redes inalámbricas tradicionales, por lo cual se deben tener consideraciones especiales al momento de implementar la seguridad, como el desgaste de la batería, el bajo alcance, poca memoria disponible, etc.

Aún con estas restricciones se sabe que no es imposible la implementación de seguridad; como se mencionó existen varias maneras de ofrecer seguridad dependiendo el tipo de sensor a utilizar, ya que no todos soportan las mismas funcionalidades.

Finalmente pudimos comprobar la carga que necesitan los sensores para los envíos, tanto en recepción como en transmisión. A partir de esto obtener y presentar que en el caso del uso de seguridad se tiene un mayor desgaste de energía el cual genera un mayor consumo de batería. Esto relacionado con los tiempos de envío necesarios.

AGRADECIMIENTOS

Este trabajo ha sido elaborado con el apoyo del CDTI, Ministerio de Industria, a través del proyecto Segur@ y a la Universidad Autónoma de Sinaloa.

REFERENCIAS

[1]. Healy, M, Newe, T y Lewis, E. Efficiently securing data on wireless sensor network. *SENSOR07*. 2007.

[2]. Gomez, Carles, Casademont, Jordi y Paradells, Josep. Theoretical Study on the Impact of Security Mechanisms on Performance of IEEE 802.15.4 and ZigBee higher layers. Barcelona : 2008.

[3]. Portal TinyOS Comunity. [En línea] http://www.tinyos.net/.

[4]. Society, IEEE Computer. Wireless Mediunm Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Network (WPAN's). New York : 2066.

[5]. Chipcon. [En línea] http://inst.eecs.berkeley.edu/~cs150/Documents/CC2420.pdf.

[6]. Rodríguez Henríquez, Francisco, y otros. Cryptographic Algorithms on Reconfigurable Hardware. : Springer, 2006.

[7]. López Trejo, Emmanuel. Implementación Eficiente en FPGA del Modo CCM usando AES. México, DF. : Septiembre 2005.

[8]. Crossbow. [En línea] http://www.xbow.com/Productos/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf.

IMPLEMENTACIÓN DE UNA APLICACIÓN SEGURA DE ENCUESTAS SOBRE TDT-MHP

Beatriz Martín de Juan, Carolina García Vázquez, Esther Moreno Martínez,
Ana Gómez Oliva y Miguel Ángel Valero Duboy.
Departamento de Ingeniería y Arquitecturas Telemáticas (EUITT)
Universidad Politécnica de Madrid
Ctra. Valencia km.7. 28031 Madrid
bmj@alumnos.upm.es, carogar@diatel.upm.es, emoreno@diatel.upm.es,
agomez@diatel.upm.es, mavalero@diatel.upm.es

***Resumen*- Este artículo detalla el trabajo de investigación y desarrollo realizado para crear un servicio interactivo y seguro en TDT que permita realizar encuestas, sondeos y consultas de opinión sobre una arquitectura cliente servidor MHP-https. La implementación de este servicio se ha realizado sobre MHP (*Multimedia Home Platform*) requiriendo que el descodificador del usuario incluya esta tecnología en su versión MHP1.1.2 pudiendo usar un canal de retorno Ethernet para enviar las respuestas. La aplicación resultante incorpora procesos de autenticación basados en certificados por lo que el escenario considerado debe usar un descodificador con lector de tarjetas inteligentes. La innovación del servicio implementado y verificado está en la incorporación efectiva sobre MHP de medidas de seguridad apropiadas (autenticación, anonimato, confidencialidad, integridad, unicidad, verificación del voto y accesibilidad) para dotar de fiabilidad a los resultados obtenidos tanto para la entidad que encarga la encuesta o sondeo como para el ciudadano que participa en ella.**

***Palabras Clave*- TDT, MHP, participación ciudadana, seguridad, tarjetas criptográficas.**

I. INTRODUCCIÓN.

La presencia actual de la televisión casi en el 100% de los hogares, unida a las nuevas posibilidades que nos ofrece la Televisión Digital Terrestre (TDT) interactiva -donde los espectadores pueden realizar la selección de las opciones mostradas en la pantalla a través del mando del televisor-, favorecerán el acceso de un público masivo a las nuevas aplicaciones que se oferten, con la ventaja adicional de que estos usuarios no requieren de conocimientos técnicos específicos o habilidades especiales. La cobertura de TDT en España alcanza ya el 93,52% de la población y su penetración del 50,7% en los hogares españoles invita a pensar en un futuro próximo de provisión de servicios interactivos cuya expansión estará condicionada por la tecnología y los modelos de negocio [1]. Todo ello brinda un marco de negocio esperanzador, aunque aún algo incierto, tanto a las empresas desarrolladoras de nuevos servicios como a aquellas entidades públicas o privadas que desean acercar sus productos a los ciudadanos, que está propiciando el desarrollo de nuevos servicios.

Sin embargo, existe un elemento a considerar en el diseño de las nuevas aplicaciones para la Sociedad de la Información que, a menudo es descuidado o su valor minimizado: la seguridad ofrecida a los usuarios y cómo es percibida por ellos. Este aspecto cobra especial importancia en servicios de comercio electrónico, e-salud, e-administración y, en general, en cualquier aplicación que conlleve un intercambio de información personal. Por esta razón, los nuevos servicios que se ofrezcan a través de la TDT, al igual que otros servicios ofrecidos a través de internet, deben poder proporcionarse bajo determinadas condiciones de seguridad que garanticen, según se requiera, la integridad de la información intercambiada, su confidencialidad, la autenticación de los comunicantes, el anonimato o el no repudio. Debido a la dificultad de su implantación, hasta la fecha son muy pocas las aplicaciones disponibles a través de TDT que incorporan estas garantías de seguridad.

Este artículo recoge los trabajos realizados dentro del grupo de investigación reconocido Sistemas Telemáticos para la Sociedad de la Información y del Conocimiento de la UPM para el desarrollo de un nuevo servicio para la TDT que permita la realización fiable de encuestas, sondeos y consultas de opinión. La particularidad del servicio implementado está en la incorporación de las medidas de seguridad necesarias para que la entidad que encarga la encuesta o sondeo pueda confiar en el valor de los resultados, puesto que tiene la garantía de que sólo los ciudadanos autorizados han podido realizarla y su opinión no ha sido alterada. Por otra parte, el ciudadano que la realiza tendrá la posibilidad de participación anónima, lo que le permite emitir su opinión sin coacción. Además, se incorpora la posibilidad de verificar los resultados en el periodo posterior a la consulta lo cual contribuirá a que los ciudadanos aumenten su confianza en el uso de este servicio.

La implementación de este servicio se ha realizado sobre MHP (*Multimedia Home Platform*) por lo que se requiere que el descodificador de TDT del usuario incorpore la tecnología MHP en su versión 1.1.2, además de incluir un canal de retorno Ethernet para que pueda enviar sus datos [2] [3]. Asimismo, el sintonizador debe incorporar un lector de tarjetas inteligentes para facilitar todos los procesos de autenticación basados en certificados.

Los trabajos recogidos en este artículo se enmarcan en el contexto del proyecto tractor SATI-TDT (Sistema de Acceso a Tarjetas Inteligentes para Televisión Digital Terrestre) financiado por el Ministerio de Industria, Turismo y Comercio, cuyo objetivo fue promover el desarrollo de nuevos servicios interactivos para la TDT. Este proyecto ha sido coordinado por *Informática El Corte Inglés, S.A.* y este grupo de investigación ha participado como entidad de I+D en tareas de diseño, implementación y validación.

II. DISEÑO DE LA APLICACIÓN.

En el diseño de la aplicación se han considerado tres escenarios, cada uno de ellos con distintos requisitos de seguridad, tanto funcionales como técnicos:

1. *Encuesta de opinión*: abarca los sondeos sobre cuestiones de importancia relativa en la que los participantes no se identifican de ninguna forma. Los usos previstos de este servicio serían la realización de encuestas de opinión, las votaciones de la audiencia sobre la programación o las votaciones en concursos.
2. *Sondeo de opinión con tarjeta/certificado*: enmarca los procesos de encuesta en las que los resultados, sin poseer validez legal, sí son lo suficientemente fiables como para ser tenidos en cuenta en la toma de ciertas decisiones. Sus principales usos podrían ser la participación de los ciudadanos en las Juntas de Distrito, seguimiento de las reuniones del Pleno de un Ayuntamiento, toma de decisiones por parte de juntas directivas de grandes asociaciones o fundaciones o la recogida telemática de firmas. En todas estas situaciones es necesario validar los datos del usuario de forma que sólo aquellos que estén autorizados puedan participar, pero garantizando siempre el anonimato de la opinión aportada.
3. *Consulta vinculante*: incorpora mecanismos adicionales que incrementan la fiabilidad. Se utilizará para las participaciones ciudadanas en las cuales los resultados tienen validez legal.

Los dos primeros escenarios tienen la suficiente fiabilidad (aunque carezcan de validez legal) como para que la entidad que lanza el sondeo pueda utilizar su resultado para tomar una decisión basada en la confianza en los datos recogidos. En cuanto al tercer escenario, posibilita la recogida telemática de firmas accesible desde el entono domiciliario para apoyar una causa determinada, lo cual supone una alternativa más a la recogida de firmas manual tradicional o a la que se puede realizar a través de internet. También se facilitará la verificación individual por parte de los participantes para lo cual se facilitarán los mecanismos necesarios, de forma que se pueda verificar la contabilización final y reclamar a una tercera parte de confianza en caso de que no se estuviera de acuerdo con la contabilización del voto. En la verificación individual será necesario almacenar cierta información en la tarjeta criptográfica del usuario, razón por la cual el DNI electrónico se ha descartado, debido a que no está permitido el almacenamiento de datos adicionales en el mismo.

El segundo escenario, *Sondeo de opinión con tarjeta/certificado*, será el principal caso de estudio en este artículo, ya que el tercer escenario está aún en fase de integración y validación.

A. Análisis de requisitos de un servicio de sondeo de opinión.

Se ha considerado que un servicio de sondeo de opinión debería cumplir los siguientes requisitos de seguridad:

- Autenticación de los participantes por medio de un censo de participantes autorizados.
- Anonimato del participante.
- Confidencialidad de los datos, tanto en su transmisión como en la recepción de los mismos, de forma que ninguna fuente no autorizada pueda conocer los resultados de la votación hasta que se cierre el plazo autorizado para votar.
- Integridad, para garantizar que la opinión no ha sido modificada durante el proceso de consulta.
- Unicidad de la opinión, para controlar que cada persona sólo participe una vez.
- Verificación del voto individual, donde el participante comprueba la validez de su voto, o global donde esto se comprueba en el recuento final.
- Accesibilidad, para que la aplicación tenga un alto grado de aceptación entre la población, independientemente de sus capacidades técnicas o físicas.

El diagrama de casos de uso de la Fig. 1 muestra los distintos actores que participan en el proceso, así como la funcionalidad del escenario que nos ocupa:

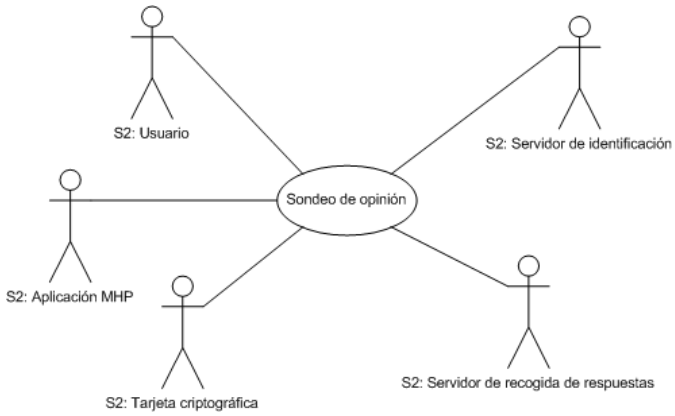


Fig. 1. Casos de uso del servicio.

En el diagrama se incluye el usuario que responde al sondeo por medio de la aplicación MHP que se ejecuta en el descodificador de TDT conectado a la televisión, *Set-Top-Box* (STB), del propio usuario. También aparece la tarjeta criptográfica, que interacciona con el usuario por medio de la aplicación segura y que le permite identificarse. Por último se muestran dos servidores: el que autentifica a los participantes y el encargado de la recogida de respuestas.

En un servicio como el desarrollado, la participación depende de la confianza de los usuarios, siendo fundamental garantizar telemáticamente el anonimato de la persona que aporta su opinión. El hecho de poder comprobar la identidad de un participante no debe ser un impedimento para el uso de la aplicación. Por esta razón, siempre buscando garantizar el anonimato de los participantes, los datos aportados por los usuarios no viajarán en ningún caso junto con su información de identificación para que pueda relacionarse una opinión determinada con un usuario concreto. El mecanismo usado para cumplir con el requisito de anonimato está basado en la firma ciega y en la existencia de dos servidores diferentes, uno que comprueba la identidad de los participantes y otro que recibe las respuestas. Hay que destacar la importancia de mantener al usuario siempre informado del funcionamiento del sistema de sondeo mediante mensajes explicativos que aportan mayor confianza en la aplicación a los participantes.

Otro de los requisitos clave en el sistema es proporcionar confidencialidad a los datos emitidos por la aplicación durante su transmisión. Con este objetivo se establece un canal seguro mediante SSL entre el STB y los servidores. Sin embargo, en este servicio es necesario además almacenar los datos cifrados en destino, de forma que nadie tenga acceso a ellos antes de que se cierre el periodo del sondeo, por tanto, además, los datos estarán cifrados en todo momento.

A la hora de participar en un sondeo o consulta es necesario poder identificar al usuario que utiliza la aplicación, de forma que sólo aquellos que estén autorizados puedan hacer uso de la misma. El proceso utilizado para la autenticación del usuario está basado en certificados digitales existentes en tarjetas criptográficas por lo que cada votante deberá poseer una tarjeta criptográfica que contenga un certificado digital a su nombre. Idealmente, esta tarjeta deberá ser el DNI electrónico, pero podría ser cualquier otra tarjeta criptográfica con certificado. El uso de una Java Card nos facilita realizar dentro de la propia tarjeta las operaciones criptográficas necesarias, tales como el cifrado y firmado de los datos, lo que aumenta la seguridad de la aplicación [4].

La unicidad es fundamental, ya que se contempla la participación sobre cuestiones de cierta relevancia, sin validez legal, donde es imprescindible garantizar que el usuario participa sólo una vez en el sondeo. El proceso de verificación de unicidad lo realiza el servidor de identificación al que se envía la información que identifica al usuario de forma anónima. Cuando el servidor reciba esta información identificativa comprueba que es válida y la contrasta con un censo para verificar si el usuario estaba autorizado o no a votar y si ya votó. Una vez más es imprescindible mantener la confianza del usuario en la privacidad de los datos que ha emitido, por lo que cobra especial relevancia el hecho de mantenerle informado de todo el proceso que se está llevando a cabo por medio de pantallas explicativas. Por último, se debe garantizar la integridad de los datos recibidos, es decir, que los datos enviados no han sido manipulados en su trayecto desde el STB al servidor. Esta integridad se garantiza mediante el firmado digital de la información que, además de autenticar el origen, garantiza la integridad del mismo.

B. Arquitectura del sistema.

La arquitectura del sistema consiste principalmente en dos servidores cuya misión es la autenticación del usuario y la recogida de respuestas y el descodificador de televisión digital (STB) MHP. A continuación se muestra el diagrama de arquitectura del sistema que se detallará posteriormente:

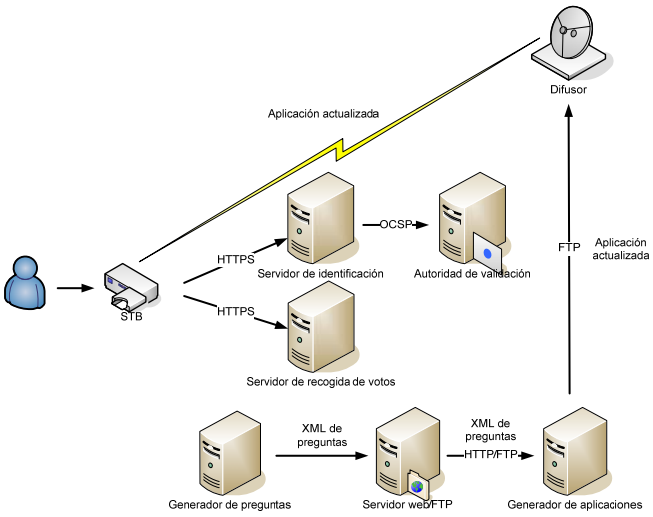


Fig. 2. Arquitectura del servicio.

En la figura anterior se puede observar la existencia de un servidor de autenticación/validación que será el encargado de validar la identidad de los participantes en el sondeo, también será el encargado de controlar que dichos participantes sólo puedan dar su opinión una única vez, para así cumplir con el requisito de unicidad. Este servidor se apoyará en una Autoridad de Validación contra la que verificará la identidad de los usuarios.

Por otro lado tenemos el servidor de recogida de datos que, como su nombre indica, será el encargado de recolectar los datos emitidos por los participantes a través de la aplicación almacenada en el STB. La comunicación entre el STB y estos servidores se protegerá mediante el protocolo SSL. Además, para garantizar la integridad de los datos se utilizará la firma digital. Por último, el requisito de confidencialidad es avalado por el cifrado de los datos.

El STB es un dispositivo con tecnología MHP 1.1.2 que dispone de canal de retorno por Ethernet y lector de tarjetas digitales incorporado. Este equipo recibirá la señal digital de televisión y además se encargará de descargar la aplicación a través de la cual el usuario participará en la encuesta.

Cada participante poseerá una tarjeta criptográfica que contiene un certificado digital a su nombre. El certificado de usuario tiene como función la identificación de los participantes frente a la autoridad de validación. La validez de los certificados se comprueba contra una autoridad de validación autorizada mediante el protocolo OCSP. Para garantizar el anonimato de los participantes en el sondeo se utiliza la firma RSA ciega que se explicará detalladamente más adelante.

La identificación y envío de datos se realiza a través del STB por medio de invocaciones SOAP a servicios web. Estas invocaciones se llevan a cabo través de un canal seguro sobre el protocolo SSL (HTTPS). El canal de retorno se establece vía Ethernet por lo que los STB deben permitir esta función, aunque también sería posible establecer el canal de retorno vía módem.

El sistema también consta de un generador de preguntas que, bajo demanda, generará las preguntas a incluir en la encuesta y las publicará en un fichero XML accesible por FTP/HTTP desde el generador de aplicaciones. La función de este fichero XML es ser descargado por el generador de aplicaciones que regenerará la aplicación MHP actualizada con las nuevas preguntas y las enviará al difusor de la señal para que vuelva a emitirla actualizada.

C. Diagrama de secuencia de mensajes.

El escenario que estamos describiendo tiene como objetivo poder aportar una mayor interactividad y seguridad en las comunicaciones entre usuarios y TDT. Para ello se han definido una serie de acciones a seguir para dotar de una mayor seguridad a los sondeos o encuestas que se realicen a través de dicha tecnología.

En la Fig. 3 se muestra el diagrama de secuencia en el que se representan las acciones llevadas a cabo por la aplicación para garantizar el funcionamiento del proceso completo de participación.

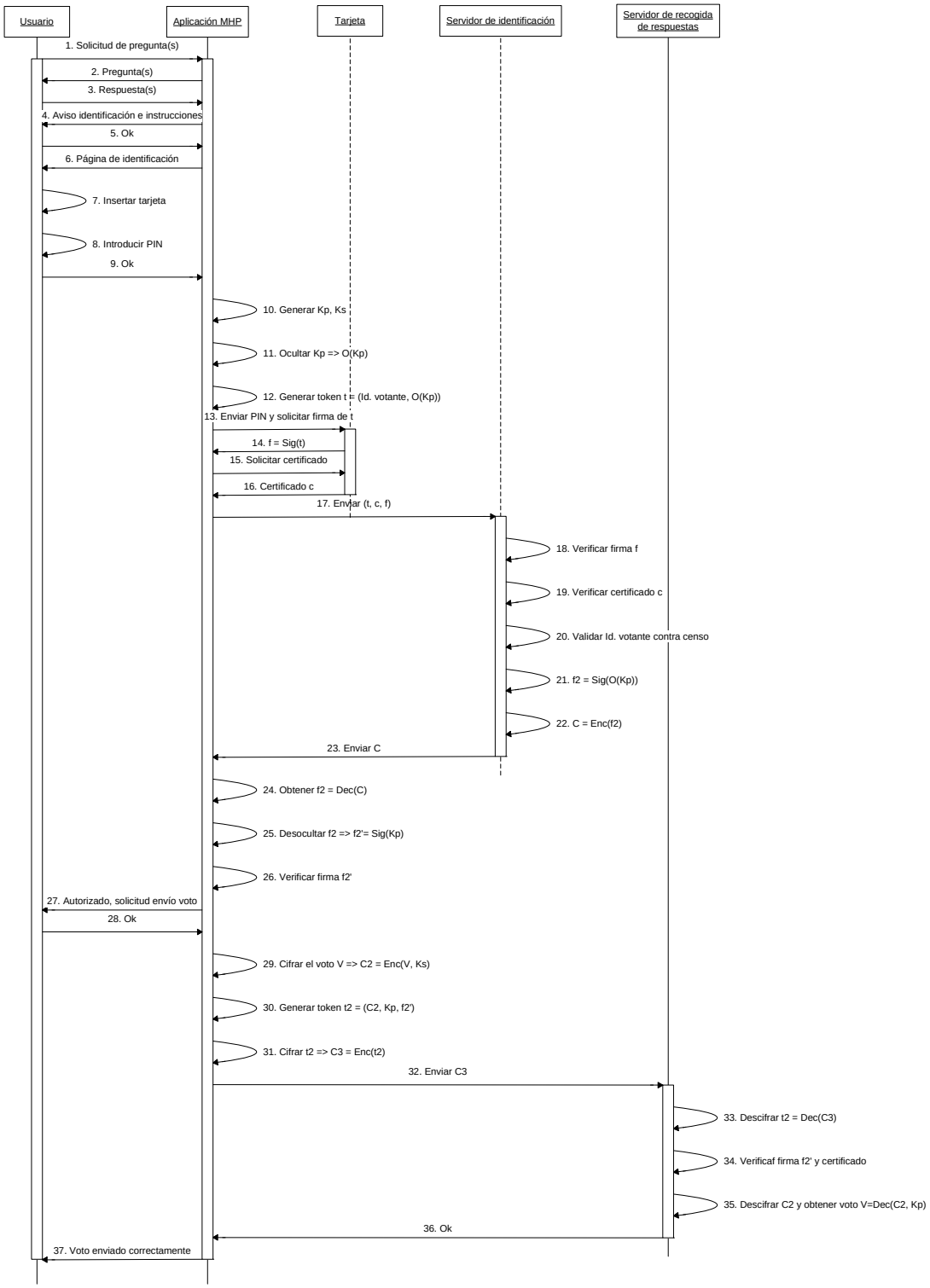


Fig. 3. Diagrama de secuencia del servicio 2.

Para realizar un sondeo de opinión a través de la TDT, en primer lugar el usuario debe aceptar la ejecución de la aplicación que maneja dicho sondeo. Una vez que se ha aceptado la ejecución la aplicación (1) mostrará una pantalla con las preguntas de que consta la encuesta y que fueron descargadas del documento XML que creó el generador de preguntas (2). Una vez que el usuario ha respondido a las preguntas (3) se procederá a la autenticación de la identidad del usuario contra el censo de participantes autorizados. El primer paso seguido por la aplicación MHP será mostrar un mensaje informativo al usuario en el que se le explicará brevemente el proceso al que se verán sometidos sus datos (4). El hecho de mantener informado en todo momento al participante la dará mayor confianza en el proceso de identificación y recolección de los datos.

Cuando el usuario acepta (5) las condiciones a las que se verán expuestos sus datos la aplicación le pedirá que introduzca su tarjeta criptográfica en el STB para proceder a la votación (6). Una vez introducida la tarjeta (7), automáticamente se mostrará una pantalla en la que se solicita el código PIN (8) con el que se validará el acceso a la tarjeta criptográfica. Todo el proceso de autenticación ante la

tarjeta ha de ser confirmado por el usuario de la aplicación para poder continuar (9).

Una vez que el usuario ha dado su consentimiento, pulsado OK, comienza el proceso de securización. Dentro de la tarjeta criptográfica, Java Card, se genera una pareja de claves de sesión RSA, pública y privada, que se emplearán para enviar los datos de forma segura y con las cuales se procederá al firmado y cifrado de la información. El hecho de crear las claves en la tarjeta criptográfica es debido a que una tarjeta es más segura que un STB y la clave privada generada, que denominaremos de ahora en adelante KSsesión, no debe ser conocida por nadie más que el usuario y por tanto no saldrá en ningún momento de la tarjeta (10).

A continuación se inicia el proceso de firma ciega RSA. Para ello se utilizará la clave pública de sesión, de ahora en adelante KPsesión, que se opacará con un factor de opacidad elegido al azar, de forma aleatoria, O(Kp) (11). Posteriormente se utilizará la KPsesión como credencial para indicar que el usuario tiene derecho a emitir su opinión en el sondeo. La clave opacada será usada por la aplicación MHP para generar un token t que es el resultado de la concatenación del identificador del votante y O(Kp) y que son los datos utilizados para la autenticación del usuario: t = (Id. Votante, O (Kp)) (12).

Una vez generado el token t la aplicación solicita a la tarjeta que lo firme digitalmente (13) con la KSsesión que se generó anteriormente. También se le solicitará a la tarjeta el certificado de usuario que contiene (15) y que como se comentó en el apartado de arquitectura es usado para autenticar al usuario frente a la autoridad de validación y que por consiguiente identifica al participante. La tarjeta devolverá tanto el certificado (16) como la firma (f=SIGusuario(t)) (14) del token a la aplicación para que ésta proceda a enviárselo al servidor de identificación. La firma nos permite avalar que la persona que emitió su opinión es quien dice ser.

La aplicación MHP envía el token t junto con el certificado c y la firma f al servidor de autenticación (17) para proceder a la verificación de la identidad del usuario:

- t es la credencial que fue opacada para que el servidor no pueda identificarla pero sí firmarla.
- c es el certificado de usuario que estaba contenido en la tarjeta criptográfica que será verificado contra la autoridad de validación por medio del protocolo OSCP (19)
- f es la firma que tiene que ser verificada en el servidor para ratificar la autenticación del usuario, una vez aplicada la KPusuario contenida en el certificado f deberá coincidir con t (18).

La única parte que seguirá oculta es la KPsesión opacada.

El servidor de autenticación valida el identificador del votante contra el censo (20) y procede a firmar O(Kp) con su clave privada f2=Sig(O(Kp)) (21). Con esta firma se finaliza el proceso de firma ciega que tiene como objetivo que un usuario pueda participar en el sondeo conservando su anonimato. La última acción por parte del servidor de identificación será devolver el resultado de cifrar f2 con la KPusuario a la aplicación para que sólo pueda ser verificada por dicha aplicación: C=ENCusuario(f2) (22-23).

La aplicación MHP remite C a la tarjeta para que obtenga f2 por medio del descifrado de C aplicando la KSusuario que se creó al principio del proceso y se lo retransmitirá a la aplicación (24). La aplicación desaplica el factor de opacidad con lo que obtiene f2'=SIGserv_ident(KPsesión) (25). Para finalizar con la autenticación del usuario y poder continuar con el envío de los datos se verifica f2' con lo que garantizamos que ha sido el servidor de identificación el que nos ha autenticado ante el censo y nos ha autorizado a seguir con el procedimiento (26). La aplicación MHP informa al usuario de que está autorizado a votar, y le solicita confirmación para finalizar con el envío de la información al servidor de recogida de datos (27).

Una vez que el usuario ha pulsado OK (28) la aplicación se comunica una vez más con la tarjeta para que proceda al cifrado de la información V con la clave privada KSsesión obteniendo como resultado: c2= ENCsesion(V,Ks) (29). Con estos datos se crea un segundo token t2 que resulta de concatenar los siguientes datos (30):

- c2 se corresponde con el voto cifrado
- KPsesión es la clave pública de sesión creada al arrancar la aplicación
- f2'=SIGserv_ident(KPsesión)

La aplicación cifra el token t2 con el certificado del servidor de recogida de respuestas (31) obteniendo los datos que se enviarán a dicho servidor: C3 = ENCserv_resp(t2) (32).

El servidor de recogida de respuestas descifra C3 con su clave privada obteniendo el token t2 en claro (33). A continuación valida la firma f2', comprobando de esa forma que el servidor de identificación autorizó la participación en el sondeo, y procede a la verificación del certificado del servidor de identificación (34). La última función de este servidor es la de descifrar los datos con la clave KPsesion (35) y si todo el procedimiento se llevó a cabo sin problemas responderá a la aplicación MHP con un OK indicando que la votación se realizó con éxito (36). Para finalizar el proceso de votación la aplicación informará al usuario, a su vez, de la recogida de la información terminó con éxito.

III. Resultados de Implementación del Servicio.

A. Entorno de desarrollo empleado.

Para la implementación final del servicio especificado en los puntos anteriores ha sido necesario utilizar diversas tecnologías.

En el caso del desarrollo de la aplicación ejecutada por el receptor TDT se ha optado por emplear MHP (*Multimedia Home Platform*), que es un middleware o soporte de intermediación entre el sistema operativo del STB y las aplicaciones superiores. Está diseñado por el proyecto DVB (*Digital Video Broadcasting*) y permite que en la capa superior se soporten aplicaciones basadas en Java, como la implementada para este servicio. Existen varias versiones de MHP, pero se ha optado por utilizar la versión 1.1.2. ya que permite aplicaciones que utilicen canal de retorno por Ethernet y la posibilidad de uso de tarjetas inteligentes basadas en Java (Java Card). Además, se pueden almacenar objetos en el STB a lo largo del tiempo, lo que se utiliza en el servicio de *Encuesta de opinión* para evitar que se supere un número de votos establecido para cada domicilio.

Para el desarrollo de la aplicación MHP se ha utilizado la herramienta de autor iDesigner [5] desarrollada por la empresa MIT-xperts. Dicha herramienta permite crear tanto la interfaz gráfica como la funcionalidad de aplicaciones interactivas basadas en Java.

La utilización de la tarjeta Java Card se justifica por la capacidad de estos dispositivos para almacenar datos y ejecutar una serie de operaciones a través de un microcomputador, formado por un único microchip que se encuentra en una tarjeta de plástico del tamaño de una tarjeta de crédito. Consta de un microprocesador que es el que realmente se encarga de dotar de inteligencia a las tarjetas. Para que dichas tarjetas puedan utilizarse han de situarse dentro o cerca de un lector de tarjetas o un dispositivo de aceptación de tarjetas (CAD). La tecnología Java Card provee de un entorno seguro para aplicaciones que se ejecutan en tarjetas inteligentes y otros dispositivos con una capacidad de memoria y procesamiento muy limitada. Permite el desarrollo de varias aplicaciones sobre una única tarjeta, aplicaciones que pueden ser añadidas posteriormente, una vez que la tarjeta ha sido entregada al usuario final por el fabricante. Las aplicaciones escritas en el lenguaje de programación Java pueden ser ejecutadas de forma segura en tarjetas de distintos fabricantes.

El entorno de desarrollo que se ha utilizado para la programación de la Java Card ha sido Sm@rtCafé® Professional Toolkit. Las características que nos aporta dicho emulador son un depurador y testeador de applets para Java Card, así como la posibilidad de crear nuevos applets y paquetes que podrán ser desarrollados de forma externa al simulador para luego cargarlos en la tarjeta de simulación. También permite un mayor desarrollo de los applets existentes y una simulación completa de la máquina virtual Java Card G&D con la emulación de todas las características de las máquinas virtuales para tarjetas inteligentes, como depuración simbólica a nivel bytecode o visualización simultánea del código fuente java y bytecode. La simulación se puede hacer paso a paso o por medio de puntos de ruptura. Por último nos permitirá la monitorización de la pila.

Para el desarrollo de las aplicaciones del lado del servidor se ha empleado J2SE (Java2 Standard Edition), lenguaje orientado a objetos desarrollado por Sun Microsystems muy utilizado en entornos de Internet. Se emplea para atender a las peticiones del cliente y para las consultas en la base de datos en el servidor. Por otro lado se han utilizado como motor el Apache Jakarta Tomcat y para la base de datos, MySQL. Ambas aplicaciones son de libre distribución.

B. Requisitos de Accesibilidad y usabilidad.

Tanto en el diseño como en la implementación de la aplicación se ha dado gran importancia a la accesibilidad y usabilidad, de forma que se facilite la participación del mayor número de usuarios posibles, incorporando así colectivos típicamente más excluidos de la Sociedad de la Información como personas mayores o con discapacidad.

Aunque hoy en día no existen unos criterios de accesibilidad claramente definidos para el ámbito de las aplicaciones de televisión digital interactiva, para el desarrollo de esta aplicación, estos parámetros se han tomado de directrices generales dictadas por la OMS, además de recomendaciones afines del WCAG 2.0 del W3C [6].

En cuanto a la accesibilidad se han contemplado varios aspectos, como el uso de colores con contraste, fuentes de texto más utilizadas por los STB del mercado con tamaños adecuados para personas con deficiencias visuales, y el uso de mensajes explicativos de utilización breves, claros y concisos que le den realimentación al usuario informándole del lugar en que se encuentra.

Por otro lado, en el ámbito de la usabilidad, se han tenido en cuenta tanto la facilidad de aprendizaje de la aplicación por el usuario como la robustez de la misma. Para ello, se ha diseñado una aplicación con navegabilidad simple e intuitiva, donde el usuario en todo momento puede salir, mostrar en pantalla completa el canal de televisión sintonizado o volver a la aplicación en el lugar que la dejó, o a la pantalla de inicio. De la navegación cabe destacar que se realiza a través del mando de la televisión, utilizando para ello las flechas del cursor y la tecla central, los botones de colores y el teclado numérico. Se ha decidido no utilizar otro tipo de teclas, como por ejemplo la del teletexto, porque en algunos receptores del mercado pueden generarse confusiones en el procesamiento de la petición realizada. Además, de esta forma, solo se utilizan los botones del mando que el usuario ya conoce.

C. Diagrama de navegación e interfaces gráficas.

En este apartado se describe la secuencia de estados e interfaces gráficas que se presentan al usuario mostrando, a su vez, ejemplos de implementación de las más relevantes. El diagrama de navegación de la aplicación es el siguiente:

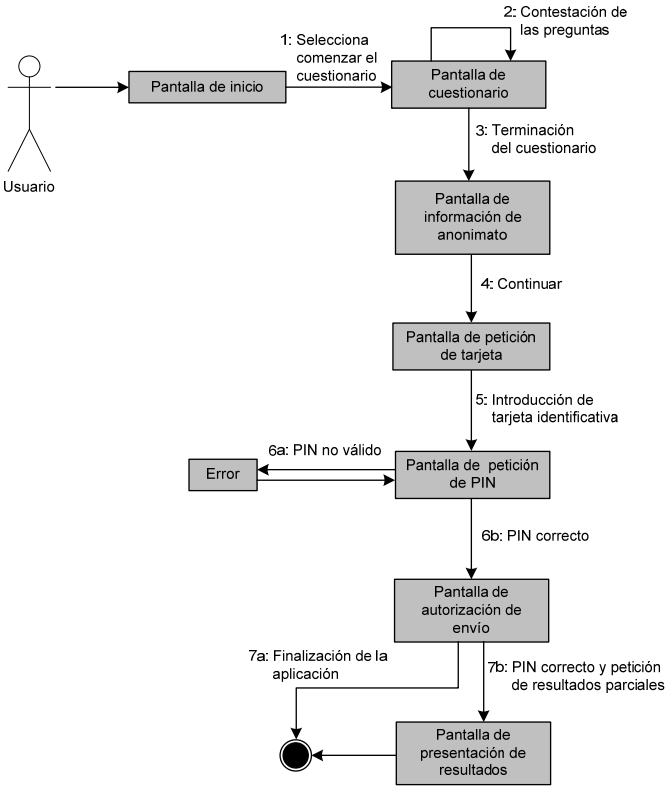


Fig. 4. Navegación de la aplicación.

- Pantalla de inicio: como su nombre indica se trata de la pantalla mediante la cual el usuario accede a la aplicación. En ella se presentan los datos básicos de la encuesta (denominación, patrocinadores, etc.) y se le da la opción al usuario de comenzar el cuestionario o salir de la aplicación.
- Pantalla de cuestionario: a través de esta pantalla, el usuario va cumplimentando la encuesta pudiendo abandonar la misma en el momento que lo desee. En la Fig. 5 se observa la realimentación que se ofrece de la navegación para garantizar la usabilidad. Las acciones que el usuario puede realizar son: salir de la aplicación, volver a comenzar el cuestionario o mostrar en pantalla completa el canal de televisión sintonizado.

Fig. 5. Pantalla de cuestionario.

- Pantalla de información de anonimato: se le muestra al participante un mensaje informativo en el que se explica, de forma clara y concisa, el proceso que se va a seguir para dejar constancia de su participación. También se le comunica de que en todo momento se mantendrá su anonimato, requisito indispensable de la aplicación.
- Pantalla de petición de tarjeta: una vez cumplimentado el cuestionario, se le pregunta al usuario si desea enviar las respuestas seleccionadas en el mismo, en caso afirmativo se le pedirá que introduzca, en el STB, su tarjeta de identificación como se observa en la Fig. 6. Si por el contrario el usuario no desea enviar nada, los datos se borran y se vuelve al inicio de la aplicación.

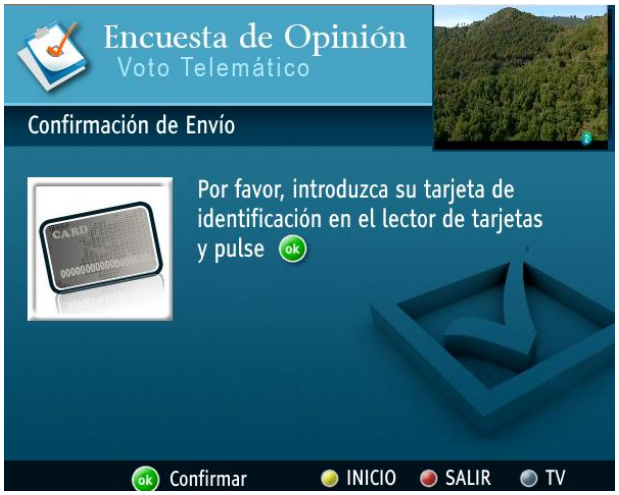


Fig. 6. Pantalla de petición de tarjeta.

- Pantalla de petición de PIN (Fig. 7): tras insertar la tarjeta y comprobar su validez en el STB, se le pide al usuario que introduzca el PIN de la misma para proceder a la autenticación en el servidor. Cuando el usuario confirma la acción, se envía el PIN al servidor, utilizando los mecanismos de seguridad explicados anteriormente.
- Pantalla de autorización de envío: tras recibir la respuesta del servidor verificando la autenticación del participante, la aplicación solicita por pantalla que autorice el envío de las respuestas dadas al cuestionario. Estas se envían de forma anónima al servidor de recogida de votos, y éste envía a la aplicación los resultados parciales, obtenidos tras la recepción de los resultados de varios participantes.

Fig. 7. Pantalla de petición de PIN.

- Pantalla de presentación de resultados: una vez recibidos los resultados parciales, la aplicación procede a mostrárselos al usuario en la pantalla de su televisión. Tras visualizarlos, el usuario puede cerrar la aplicación o volver al inicio de la misma.
- Pantalla de error: se muestra cuando se produce algún error en el transcurso de la aplicación, como por ejemplo, la introducción de una tarjeta no válida o de un PIN erróneo, fallo en la conexión con el servidor, etc.

D. Verificación de la solución desarrollada.

El principal objetivo propuesto fue el de crear un sistema seguro de votación electrónica para el sondeo de opinión basado en el uso de certificados digitales y tarjetas inteligentes. Los principales requisitos exigidos para la consecución de este objetivo, y que ya fueron nombrados anteriormente, son el anonimato, la confidencialidad, la autenticación, la unicidad, la integridad y la accesibilidad, los cuales se han cumplido satisfactoriamente.

El uso de la firma ciega nos garantiza el anonimato del votante. Su uso se apoya en dos servidores: el de recogida de respuestas y el de identificación. La autenticación se basa en el uso del identificador de usuario, que nunca viajará junto con el voto. La autenticación del usuario se verifica en el servidor de identificación a partir de un censo de votantes autorizados. La unicidad del voto se garantiza contrastando los datos del votante con el censo, de forma que, una vez comprobado que está autorizado, si ya ha votado se impide que vuelva a votar.

La integridad y autenticidad de un usuario se basan en el uso de la firma digital. Esta firma se lleva a cabo en el interior de la tarjeta criptográfica para evitar el uso de la misma por alguien que no sea su dueño o conozca el PIN.

El servicio de confidencialidad e integridad de la información se lleva a cabo, una vez más, en el interior de la tarjeta criptográfica y es proporcionado por medio del cifrado de la información. El voto emitido por el usuario se cifrará con la clave privada de usuario almacenada en la Java Card.

Por último, se cumplen los requisitos de accesibilidad y usabilidad, ya que se han tenido en cuenta el contraste de los colores, tamaño de letra adecuada, realimentación y facilidad en la navegación. Al acceder a la aplicación mediante el mando de la televisión se garantiza la accesibilidad, ya que es la tecnología con mayor tasa de penetración entre la población.

IV. Conclusiones.

La televisión es una de las tecnologías con mayor tasa de penetración actualmente lo cual será un hecho también creciente en el ámbito de la TDT. El surgimiento de la televisión digital interactiva, junto con la mayor penetración de la banda ancha en el hogar, hace que se haya convertido en un potente medio de acceso a nuevos servicios interactivos, con la ventaja de que no es necesario disponer de conocimientos técnicos específicos para utilizarlos. Hoy en día existen proyectos dedicados a suministrar ofertas de aplicaciones de este tipo, como por ejemplo el ayuntamiento de Alcázar de San Juan (Ciudad Real), la que ha sido la primera "isla digital" de la TDT en España. En este prototipo de ciudad digital se ofrecen servicios tanto informativos como interactivos o transaccionales, la mayoría relacionados con la denominada t-administración, aunque hasta el momento no se contempla el uso de tarjetas criptográficas para securizar los servicios ofertados [7].

La aplicación diseñada, integrada y validada en este trabajo de investigación y desarrollo permite emplear de modo eficiente mecanismos de seguridad sobre TDT-MHP y HTTPS basados en el acceso a tarjetas criptográficas para aumentar la confianza en los dos lados de la comunicación y poder realizar encuestas masivas a la población. El rendimiento operativo de esta arquitectura es suficiente para los requisitos funcionales y de usabilidad esperados por el ciudadano no observando tiempos de retardo o cifrado críticos para un acceso apropiado a los servicios de participación ciudadana. Asimismo, las interfaces y arquitectura planteada fortalece la confianza por el lado de los participantes ya que se ofrecen mayores garantías de que se mantendrá su anonimato y que su voto no será modificado en el transcurso del sondeo, y por el lado de la empresa u organización, con la certeza de que los datos, pese a no poseer validez legal en el escenario mostrado anteriormente, sí tienen la suficiente fiabilidad como para tomar una decisión a partir de los mismos. Los trabajos desarrollados han sido validados extensamente en descodificadores disponibles en el mercado asegurando su validez para un modelo cliente servidor como el planteado. En los trabajos actuales se propone como futura mejora la inclusión de las aplicaciones en una maqueta completa de DVB que permita medir efectivamente los tiempos de respuesta en la cadena de distribución de la aplicación y acceso al servidor completa.

Agradecimientos.

Programa Nacional de Tecnología Electrónica y de Comunicaciones / Acción Estratégica sobre Televisión y Radio Digital, Secretaría de Estado de Telecomunicaciones y para la Sociedad de la Información, Ministerio de Industria, Turismo y Comercio.

Javier Herrero, David Huerta y Carlos Garcés. Informática El Corte Inglés, S.A.

Referencias.

[1] Asociación para la Implantación y el Desarrollo de la Televisión Digital Terrestre en España (impulsatdt), *Informe mensual N. 27:* abril 2009.

[2] European Broadcasting Union, *Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.1.2*, Rev. 1, DVB Bluebook A068: 2005. http://www.mhp.org/specs/a068r1_mhp_112.zip

[3] Piesing, J., *The DVB Multimedia Home Platform (MHP) and Related Specifications*, Proceedings of the IEEE , v. 94, no.1, pp.237-247: 2006

[4] Sun Microsystems, *Java Card Technology*: 2003. http://java.sun.com/javacard/index.jsp

[5] Mit-xperts, *Products: iDesigner*, MIT-xperts GmbH, Munich: consultado en marzo de 2009. http://www.mit-xperts.com/products/

[6] Word Health Organization, *What is E-Accessibility?*, Online Q&A: diciembre de 2008. http://www.who.int/features/qa/50/en/

[7] J. A. Sánchez, *Proyecto Alcázar Digital*, Ayuntamiento de Alcázar de San Juan: 2008. http://www.alcazardigitaltdt.com/

[8] Giesecke & Sm@rtCafé® Expert 3.x. www.gi-de.com/pls/portal/maia.display_custom_items.DOWNLOAD_FILE_BLOB?p_ID=88534. Consultado en abril 2009

Janus: un Generador de la Vista de Roma Framework basado en Plantillas

Pablo Martín, Guillermo Hernández
Division I+D+i
Informática Gesfor (Grupo Gesfor)
Avda Manoteras, 32 28050 Madrid
{pmartinb,ghernandezc}@grupogesfor.com

Carlos A. Iglesias
Division I+D+i
Germinus XXI (Grupo Gesfor)
Avda Manoteras, 32 28050 Madrid
cif@germinus.com

Luca Garulli, Giordano Maestro
Division I+D+i
Asset Data s.r.l.
Via Rhodesia, 34 00144 Roma
{luca.garulli, giordano.maestro}@assetdata.it

***Resumen*—Los enfoques tradicionales de desarrollo de aplicaciones web han mostrado serios problemas de productividad, lo que está dando lugar a nueva generación de frameworks web que automatizan en gran medida el desarrollo para aplicaciones principalmente de persistencia de datos. En este proyecto se presenta cómo la utlización de plantillas puede facilitar la generación de la capa de presentación en uno de estos nuevos frameworks ágiles, Roma Metaframework. Este trabajo ha sido desarrollado dentro del proyecto ICT Romulus. La solución, llamada Janus, utiliza un mecanismo de plantillas basado en FreeMarker para extraer información de los objetos que definen la vista de la aplicación generando una estructura de páginas JSP equivalente, y se comunica con el framework mediante una serie de etiquetas Java. Como resultado, Janus permite la generación automática de una interfaz gráfica personalizable, implementada con JSP, CSS y Javascript, con un tiempo de desarrollo reducido.**

***Palabras Clave*—Janus, JSP, Roma, Metaframework, Java, Javascript, FreeMarker, Romulus**

I. Introducción

El Desarrollo Web es una de las áreas más activas, tanto a nivel nacional, como europeo, pero también es un área poco madura debido a la proliferación de frameworks y componentes. En concreto, Java Enterprise Edition es la opción preferida por los europeos con más del 38.6 % de desarrolladores usuarios de esta tenología según los datos de Evans [5]. Este hecho ha provocado que las habilidades requeridas por los desarrolladores web para llevar a cabo un desarrollo software, aumenten, y con ello baje su productividad y la fiabilidad de sus desarrollos. Este problema es particularmente acentuado en el entorno java, en el que el desarrollo de aplicaciones exige multitud de conocimientos de frameworks y tecnologías (Struts, Hibernate, Java Server Pages, XSLT, JUnit, ...). En este contexto surge el proyecto Romulus [18], cuyo principal objetivo es investigar y desarrollar nuevos métodos para aumentar la productividad y la fiabilidad del desarrollo de aplicaciones web en Java.

El resto del artículo se estructura como sigue. La sección II hace una breve introducción al proyecto Romulus donde se encuadra Janus. La sección III presenta las necesidades que se pretenden cubrir y las deficiencias que presenta Echo2, seguida de la sección IV donde se explica la implementación de Janiculum como mecanismo actual para mostrar la vista de las aplicaciones en Roma. A continuación, en la sección V, se exponen las herramientas de generación de plantillas más populares y se define la arquitectura de la solución propuesta en la sección VI. La sección VII ilustra con un ejemplo el funcionamiento de esta solución y, por último, se recogen las conclusiones y el trabajo futuro en la sección VIII.

II. El proyecto Romulus

El proyecto ROMULUS [18] tiene el objetivo de contribuir en en el desarrollo de tecnologías ágiles para el desarrollo de aplicaciones web en Java. El proyecto investiga en diseño dirigido por el dominio (DDD, *Domain Driven Design*) para aplicaciones web basado en el metaframework Roma, así como en la integración de la tecnología de mashups en el ciclo de vida de desarrollo software.

En el proyecto participa la gran empresa, **Informática Gesfor**, como coordinadora, las PYMEs **Asset Data**, **Liferay**, **IMola** e **ICI** y los centros de investigación: **Universidad Politécnica de Madrid** y el instituto **DERI**.

Romulus propone la mejora de la productividad del desarrollo de las aplicaciones web, así como de su calidad, mediante cuatro objetivos:

- Diseño Dirigido por el Dominio basado en un Metaframework. Como se presenta en la sección II-A, el proyecto investiga en el uso de un metaframework y en centrar los esfuerzos en el modelado del dominio, empleando técnicas de generación automática de código para los frameworks Java más populares.
- Desarrollo Orientado a Mashups. Reutilización de componentes y servicios disponibles para el desarrollo y extensión de aplicaciones.
- Introducción de objetivos no funcionales (seguridad, fiabilidad, análisis de prestaciones) en todo el ciclo de vida.
- Investigar en el balance adecuado entre tecnologías cliente, tecnologías servidor, y tecnologías de scripting. El proyecto investiga en la definición de buenas prácticas y patrones de diseño para repartir adecuadamente la carga entre estos puntos.

Estos objetivos suponen, en términos de productividad, una **reducción del 20 % en tiempos de desarrollo** de aplicaciones web estándar y un **30 % menos de conocimientos requeridos**, basándose en el número de librerías necesarias y su complejidad para implementar una serie de tareas estándar [17].

II-A. Roma Metaframework

Roma [6] propone un nuevo paradigma para el desarrollo de aplicaciones web tomando ventajas de las nuevas tendencias en ingeniería de software tales como el diseño

dirigido por dominio (*Domain Driven Design*) combinado con metodologías de desarrollo ágil y algunos de los principios de Ruby on Rails [16]. Para lograr este objetivo funciona como un meta-framework que permite desarrollar aplicaciones utilizando cualquier framework como una pieza intercambiable, desacoplada de la aplicación en sí. Esto permite la integración de las nuevas tecnologías sin modificar el dominio ni la lógica de la aplicación y también aumentar las tasas de productividad en el desarrollo de software.

Roma propone tres niveles de abstracción:

- El *Dominio*, definido por el desarrollador y que tiene que ver con el modelo de negocio de la aplicación.
- Los *Aspectos*, implementados por el framework (persistencia, vista, internacionalización, sesión, monitoreo, flujo, etc).
- Los *Módulos*, que proporcionan facilidades para acelerar el desarrollo, promoviendo la reutilización *de* código (persistencia, login, gestión de usuarios, integración con IDEs, etc.).

Las características principales de Roma son:

- Roma está basada totalmente en POJOs (Plain Old Java Objects)
- Todos los aspectos son orientados a objetos: desde el modelo a la vista.
- Promueve el uso del diseño dirigido por dominio (Domain Driven Design).
- Funciona con convenciones del estilo de Ruby On Rails: mucho menos código que escribir y mantener y mayor uniformidad de los proyectos.
- Las aplicaciones son portables a otros frameworks gracias a la orientación a POJOs.
- Permite modificar directamente el framework, se pueden desarrollar módulos adicionales o extender los ya existentes.
- Está basado en Spring Framework pero se pueden extender otros frameworks.

Roma proporciona al usuario soporte automático en cada capa y aspecto de su aplicación, desde interfaces web de usuario dinámicas y persistencia, hasta funcionalidad de informes, desarrollo de portlets y tecnologías semánticas. Proporciona interfaces de comportamiento muy genérico llamadas “Aspectos”. Los Aspectos incluyen los casos de uso más comunes, y permiten hacer independiente el código de la tecnología, con lo que en caso de futuras migraciones de tecnología, el proceso será más rápido y sencillo. La figura 1 muestra los aspectos implementados por Roma. En este artículo se describirá el módulo Janiculum, que implementa el aspecto View, y se presentará Janus como una alternativa para la vista de las aplicaciones.

III. Necesidades

La interfaz gráfica de las aplicaciones es un aspecto complejo que, siguiendo la filosofía de Roma, debe ser independiente de la lógica de la aplicación y, además, debe permitir el uso de diferentes tecnologías subyacentes. Roma proporciona por defecto una interfaz gráfica para las aplicaciones generada con la tecnología de Echo2. Echo2 es una plataforma basada en AJAX que ofrece una capa de Java que genera dinámicamente Javascript. Con esto permite la construcción de aplicaciones

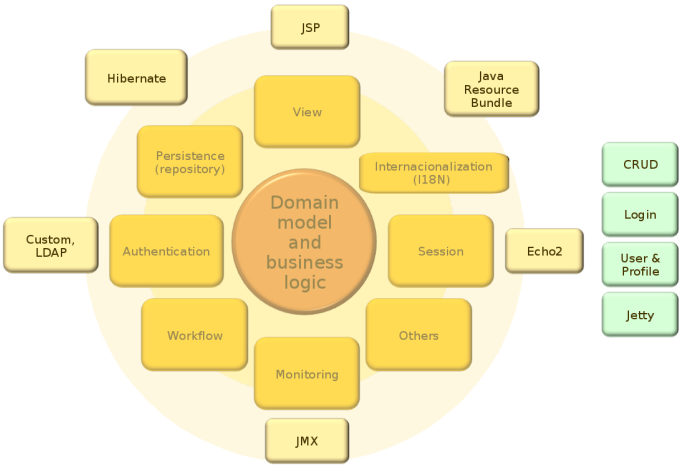


Figura 1. Aspectos de Roma

web aprovechando las capacidades de los clientes ricos. Las aplicaciones se realizan enteramente en Java mediante una API orientada a componentes y dirigida por eventos [13]. La ventaja que proporciona esta solución es su potencia y la total abstracción de la capa web. Sin embargo tiene una serie de deficiencias que motivan el desarrollo de otra solución para generar la vista de las aplicaciones web en Roma. La primera de sus limitaciones es la gestión de eventos. Al ejecutarse sobre un servidor, los eventos sólo se envían desde un cliente bajo ciertas circunstancias, por ejemplo, al hacer clic sobre un botón. Esto hace que para ciertas operaciones, como por ejemplo autocompletar o desplazar el ratón sobre un elemento, sea necesario crear componentes personalizados, lo cual influye negativamente en la productividad de la aplicación. Además Echo2 genera identificadores, dentro del código Javascript, que cambian con cada sesión lo que hace que dicho código generado no se pueda modificar. Otro problema de Echo2 es la generación de URLs ya que sus aplicaciones sólo cuentan con una URL global lo que resulta negativo para posicionarse en buscadores. Finalmente, el uso de esta plataforma requiere de una formación adicional ya que no se trata de un framework muy extendido y, además, su comunidad está decreciendo año tras año. Todas estas limitaciones hacen necesaria la utilización de otra tecnología más flexible en la personalización y que esté ampliamente difundida para mejorar la presentación y aumentar la productividad. La solución que presenta Janus soluciona los problemas de Echo2 ya que utiliza tecnologías de CSS y JSP, muy extendidas en el desarrollo de interfaces web, con una gran cantidad de librerías disponibles e independientes de servidor y plataforma. Janus genera una interfaz web en JSP que el desarrollador podrá utilizar directamente o modificar según sus necesidades. Esta solución permite que la aplicación presente diferentes URLs ya que se genera una estructura de páginas JSP similar a la que se genera en frameworks semejantes como Ruby on Rails o Grails donde se utilizan mecanismos de plantillas para generar la interfaz gráfica de las aplicaciones.

IV. Modulo Janiculum de Roma

Para solucionar algunos de los problemas de Echo2, dentro de Roma se ha desarrollado un aspecto llamado Janiculum

que traslada los conceptos de Roma a páginas dinámicas utilizando standards XHTML y CSS2. Para ello tiene que realizar una correspondencia entre los objetos de Java y las áreas definidas en el layout de Roma. Posteriormente se realiza una asociación entre los elementos de la vista y los objetos. Para ello Janiculum realiza las siguientes operaciones:

- Generar la estructura de la página a partir de la definición de las áreas de pantalla
- Generar fragmentos HTML para cada Objeto/atributo/método
- Asociar identificadores únicos para cada Objeto/atributo/método
- Realizar la conversión entre los parámetros de la petición (Strings) y los valores del atributo del objeto
- Identificar qué acción se ha realizado en el cliente e invocar el método correspondiente en el objeto ubicado en el servidor

IV-A. Definición de áreas en Roma

Roma permite definir áreas donde colocar los diferentes elementos de los POJOs, independientemente de la implementación utilizada para la vista. Estas áreas se definen en ficheros XML que se colocan por defecto dentro del paquete view.domain.screen, donde se encuentra la definición por defecto **main-screen.xml** (figura 2):

Este archivo permite definir las áreas y su colocación dentro de la pantalla [12]. La colocación de los elementos dentro de dichas áreas se realiza mediante anotaciones con el parámetro *layout*. Por ejemplo la anotación @ViewClass(layout = ”screen://body”) para una clase colocará dicha clase dentro del area llamada *body*. Esto permite definir la colocación de los elementos desde las clases java, sin hacer referencia a ninguna tecnología, lo que permite cambiar de Echo2 a JSP u otra tecnología sin necesidad de modificar la aplicación.

```
<?xml version="1.0" encoding="UTF-8"?>
<screen xmlns="http://www.romaframework.org
/xml/roma"
        xmlns:xsd="http://www.w3.org/2001
/XMLSchema-instance"
        xsd:schemaLocation="http://www
.romaframework.org/xml/roma http://www
.romaframework.org/schema/roma-view-screen
.xsd">

        <area name="main" type="grid" size="1">
        <area name="main" type="column">
                <area name="header" />
                <area name="menu" />
                <area name="body" />
                <area name="footer" />
        </area>
</screen>
```

Figura 2. Definición por defecto de las areas en main-screen.xml

IV-B. Taglib de Roma

Como se ha descrito anteriormente Roma proporciona una librería de etiquetas o taglib (**roma.tld**) que permite la comunicación entre la vista y el controlador generando código con los identificadores unívocos que utilizará internamente Roma. Esta taglib dispone de tres etiquetas básicas:

- *<roma:class>* - Muestra el objeto completo utilizando la renderización por defecto.
- *<roma:field name=”fieldName”part=”partName”>* - Muestra un atributo del objeto. El campo name (obligatorio) debe contener el nombre de dicho atributo mientras que part (opcional) permite especificar cómo se mostrará el atributo: entero, sólo la etiqueta o sólo el contenido.
- *<roma:action name=”actionName”>* - Muestra un método del objeto. El campo name (obligatorio) debe contener el nombre de dicho método. Por ejemplo, con la etiqueta: *<roma:field name=”address” />*, el taglib generará el código HTML correspondiente al atributo address del objeto que se esté mostrando.

Con esta implementación Roma proporciona una interfaz gráfica basada en HTML que genera páginas dinámicamente a partir de los POJOs. Si un usuario quiere modificar el aspecto de un determinado POJO (Ejemplo.java) puede crear el archivo Ejemplo.jsp y personalizar su vista utilizando las etiquetas proporcionadas por la taglib de Roma. Esto resulta poco productivo porque para personalizar el aspecto de la aplicación sería necesario crear manualmente el archivo correspondientes a cada POJO. Por ello se ha implementado un mecanismo de plantillas llamado **Janus** como una alternativa que genere todos estos archivos automáticamente y que, además, permita personalizar fácilmente y desde un único punto la forma en que los archivos se generan.

V. HERRAMIENTAS DE GENERACIÓN DE PLANTILLAS

En esta sección se presenta un estudio de las distintas herramientas existentes para la generación de plantillas. Una plantilla [23] es una forma de dispositivo que proporciona una separación entre la forma o estructura y el contenido. Es un instrumento que permite guiar, portar o construir un diseño o esquema predefinido. Hay una gran variedad de herramientas para generar plantillas, como por ejemplo: Beilpuz [3], eRuby [7], Dwoo [19], Evoque Templating [20], GvTags [8], h2o [9], etc.

Velocity [1] es un generador de plantillas basado en Java y de sintaxis similar a Javascript. Permite a los diseñadores web referenciar métodos definidos en código Java. Esto proporciona a la página web mayor mantenibilidad, ya que separa el código del diseño, y es una alternativa viable a JSPs o PHP.

FreeMarker [15] es una librería Java que proporciona un generador de plantillas para generar texto. Ha sido diseñado para su uso por aplicaciones basadas en servlets y en el modelo MVC (Modelo Vista Controlador) para la generación de páginas web HTML. FreeMarker no es un lenguaje de programación, simplemente genera texto a partir de programas java y lo muestra usando plantillas.

Si comparamos ambas soluciones, la primera conclusión que se puede extraer es que Velocity es una herramienta *más simple* y ligera pero con un lenguaje de plantillas menos potente que no permite tantas operaciones como FreeMarker [14]. Además Velocity hace un uso directo de los métodos de los objetos java y mueve tareas de presentación al código del controlador lo que contradice los principios del patrón MVC. FreeMarker es más estricto con esto, lo que puede generar problemas con algunas operaciones, pero el html

generado es más correcto. La mayor ventaja de Velocity frente a FreeMarker es su comunidad mucho mayor y el soporte de Apache. Además, al ser más simple, es también más rápido que FreeMarker en algunas operaciones.

Sin embargo FreeMarker ofrece una serie de características que no pueden ser realizadas con Velocity de forma trivial. Las más importantes de estas características son las mejores macros, la capacidad de utilizar las bibliotecas de etiquetas JSP personalizadas en plantillas o trabajar directamente sobre objetos de Python y el uso de namespaces.

Con FreeMarker se puede convertir también plantillas de Velocity a plantillas de FreeMarker gracias a una herramienta propia [22]. Ambos mecanismos de plantillas son similares para las operaciones básicas y tienen buen rendimiento con plantillas pequeñas, si bien un factor a tener en cuenta es que FreeMarker es más rápido a la hora de procesar plantillas grandes. Por tanto, la potencia de FreeMarker y su elegancia lo convierten en una mejor opción para su integración con Roma.

VI. Definición de la arquitectura y solución

El componente Janus genera páginas JSP a partir de los objetos java de la aplicación y utiliza tecnología de CSS2 para la apariencia. Para la integración con el metaframework de Roma se utilizan anotaciones java [21] y las etiquetas proporcionadas por el taglib de Roma (**roma.tld**). Para esta generación de código, Janus utiliza **FreeMarker** como mecanismo de plantillas e integra tecnología de Javascript para mejorar la presentación, mediante la utilización de **JQuery UI** [2]. La Figura 3 muestra la estructura básica del módulo y el flujo de información para generar el código jsp final.

VI-A. Definición de Plantillas Dinámicas para Roma

Este apartado recoge la definición de plantillas dinámicas para Roma basadas en FreeMarker que facilitan la adaptación de las vistas de una aplicación basada en el metaframework Roma. Gracias a la integración de FreeMarker con Java, se ha realizado una biblioteca de etiquetas o taglib de Roma que permite la comunicación de la vista con el modelo, lo que posibilita la generación de diferentes vistas cambiando únicamente estas plantillas de FreeMarker.

La implementación por defecto de la plantilla de Janus para Romulus organiza el código de los archivos JSP generados colocando en una sección los atributos del objeto, cada uno dentro de un . A continuación, en una nueva sección, se colocan los métodos del objeto. La figura 4 muestra un fragmento de esta plantilla con la generación de etiquetas para los métodos y atributos del objeto utilizando la taglib de Roma.

```
...
<!-- fields of the pojo -->
<div class="${classname}_fields">
<#list fields as fieldname>
   <div>
        <roma:field
        name="${fieldname}" />
   </div>
</#list>
</div>

...

<!-- actions of the pojo -->
<h2>${classname} actions</h2>
<div class="buttonBar">
        <#list actions as actionname>
                <roma:action
                name="${actionname}"/>
        </#list>
</div>
...
```

Figura 4. Fragmento de la plantilla por defecto de Janus para Romulus

El código resultante generado con esta plantilla presentará los atributos y los métodos en diferentes columnas como muestra la figura 5.

VI-B. Configuración de Plantillas Dinámicas para Roma

Una plantilla de Romulus permite los siguientes constructores:

- *Date*. Las fechas se muestran con el calendario JQuery UI Datepicker.
- *ContainerPage*. Esta clase de Roma se muestra utilizando los tabs de JQuery UI.

Esto se puede configurar mediante el archivo *constructors.xml* que permite definir las asociaciones entre constructores y elementos de JQuery UI.

Además de esta estructura de los JSPs generados, válida para cualquier objeto, Janus tiene un comportamiento específico para las clases que Roma genera automáticamente

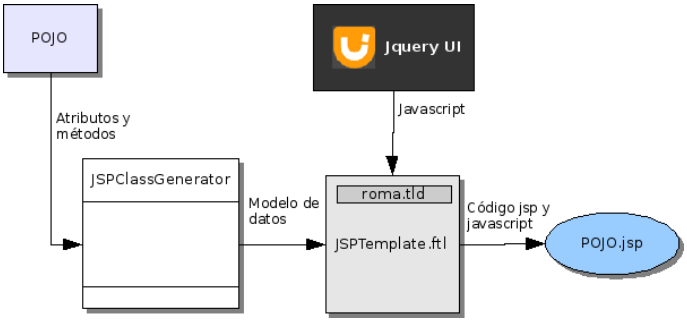


Figura 3. Flujo de información de Janus

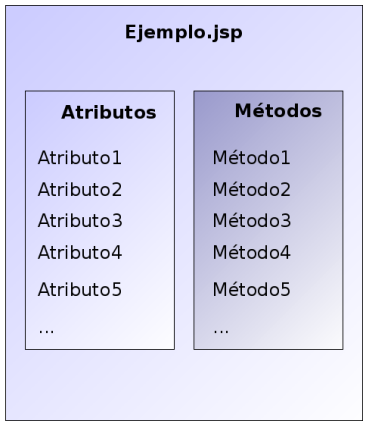


Figura 5. Estructura de los JSPs generados con la plantilla de Romulus

para facilitar las tareas de creación, lectura, actualización y borrado (CRUD). Para estas clases la generación de código está optimizado para una mejor usabilidad mostrando un filtro de búsqueda en la columna izquierda y los resultados, junto con las operaciones CRUD en la de la derecha como muestra la figura 6.

En el futuro se podrá definir una biblioteca de elementos para el cliente (Javascript, CSS, ...) de forma que se puedan establecer fácilmente correspondencias entre estos elementos de la biblioteca y los elementos del POJO. Estas correspondencias servirán para definir perfiles que permitirán a la aplicación ofrecer diferentes vistas dependiendo del perfil del usuario.

VI-C. Estructura y funcionamiento

Para conseguir la independencia de la vista en las aplicaciones en Roma se define la interfaz a través de objetos Java (POJOs) con anotaciones localizados dentro de un paquete específico llamado *view* [11]. El componente Janus crea una estructura de archivos JSP paralela a la existente en dicho paquete de forma que cada objeto java de ese paquete (o bien aquellos que el usuario elija) tengan un archivo JSP correspondiente. Esta estructura de archivos JSP tiene una jerarquía al igual que los objetos java de forma que el nivel más alto está representado por la clase Object.jsp que Roma provee por defecto. Estos archivos jsp se colocan en un directorio dentro de la estructura de directorios que Roma crea para cada aplicación. La Figura 7 muestra esta estructura.

Dentro del directorio *dynamic* se encuentran las hojas de estilo y los archivos jsp correspondientes a los objetos Java. En el directorio *static* se encuentran los archivos jsp, hojas de estilo y librerías correspondientes a las tecnologías externas que se quieran integrar en el proyecto. En este caso se trata de la librería de Javascript JQuery UI [2] y la librería de CSS Yaml [10].

El comportamiento de Roma para mostrar un objeto consiste en buscar un archivo jsp cuyo nombre sea igual al nombre de la clase de dicho objeto acabado en .jsp. Si existe, Roma utilizará este archivo para mostrar el objeto y, si no, irá ascendiendo por la estructura jerárquica hasta hallar el jsp adecuado o llegar a Object.jsp quien mostrará el objeto de la forma definida por defecto.

El componente Janus crea archivos jsp a partir de objetos java de forma que se puedan modificar fácilmente sin necesidad de compilar el código fuente de la aplicación. Para ello hace uso del taglib de Roma que genera el código html adecuado para los atributos y métodos del objeto que se va a mostrar, con los identificadores unívocos que Roma podrá utilizar para su funcionamiento interno. Este taglib se incluye en los archivos JSP que se generan con Janus de forma que para generar el código correspondiente a los atributos y los métodos basta con poner la etiqueta correspondiente del taglib dentro del archivo JSP generado.

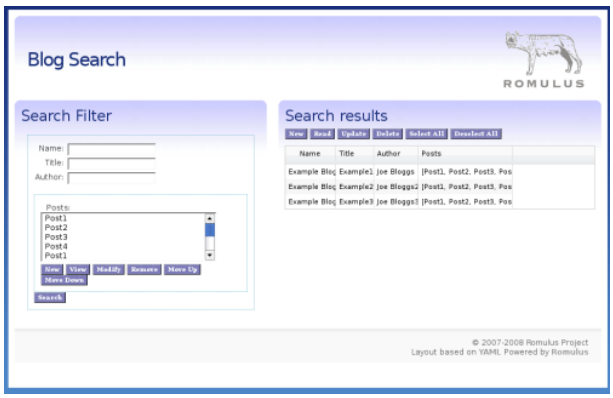


Figura 6. Estructura de los JSPs para las clases CRUD de Roma

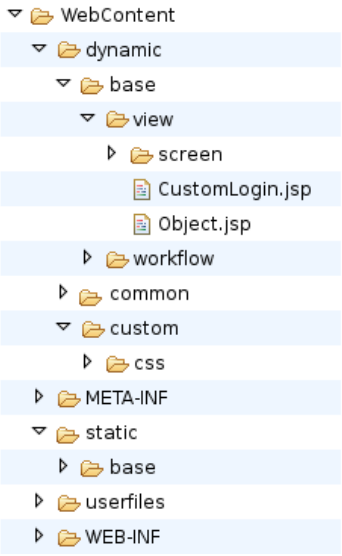


Figura 7. Estructura de directorios para los archivos JSP

El funcionamiento de Janus para generar código JSP es el siguiente: mediante introspección se recorre el objeto que se desea mostrar añadiendo al modelo de datos de FreeMarker todos sus métodos y atributos. Posteriormente la plantilla JSPTemplate.ftl accede a este modelo de datos organizando los atributos y los métodos con las etiquetas correspondientes del taglib de Roma: **roma.tld**. La ventaja de este comportamiento es que el usuario puede modificar la plantilla consiguiendo cambiar la forma en que se generan los archivos JSP finales. Concretamente, se pueden integrar librerías de Javascript para mejorar la apariencia de los JSPs generados.

Para la implementación del Janus se ha incluido tecnología de JQuery UI [2] para organizar los atributos y los métodos en diferentes pestañas, así como para el renderizado de algunos tipos de datos como por ejemplo los atributos de tipo Date que son mostrados utilizando el calendario JQuery UI Datepicker. Además se ha incluido tecnología de Yaml [10] para el diseño de las columnas.

Esta solución proporciona flexibilidad para la implementación de la interfaz gráfica por parte de los desarrolladores ya que pueden retocar individualmente los jsp's generados mientras que se reduce el tiempo de desarrollo al permitir modificar el código de todos los archivos jsp desde un único punto y sin necesidad de compilar. La siguiente sección muestra un ejemplo del uso de Janus para crear la interfaz de una aplicación.

VII. Ejemplo de aplicación o futuras aplicaciones

El proyecto Scrooge [4] es un proyecto que se utilizará como demostrador del proyecto Romulus y que esta desarrollado en parte con Romaframework.

El esqueleto de la aplicación está creado con Romaframework. Como se ha explicado anteriormente, la tecnología Echo2 es muy poco flexible para realizar cualquier cambio en el estilo de las aplicaciones, al estar escrita la interfaz directamente en código java, de modo que se ha utilizado el módulo Janus para generar una vista en JSP, a partir de cada objeto, mucho más flexible, con lo que se permite mayor personalización a los diseñadores web, ya que pueden utilizar JQuery para conseguir aplicaciones web más vistosas. La figura 8 muestra el aspecto de la interfaz de Scrooge.

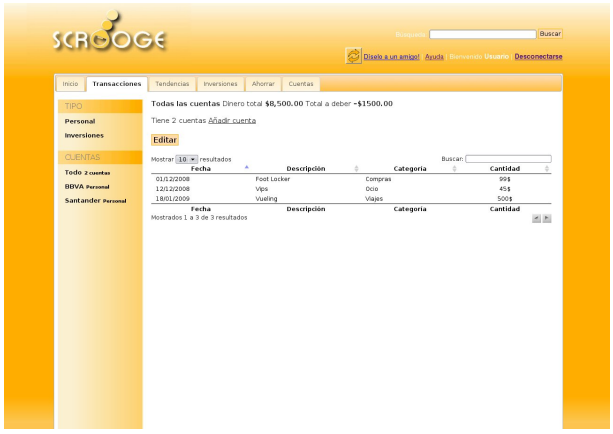


Figura 8. Interfaz de Scrooge, desarrollada con Janus

El proyecto Scrooge, aun en fase de desarrollo, hace uso de tecnologías web 2.0, AJAX y JQuery por lo que utilizará todas las facilidades aportadas por Janus.

La solución Janus ha sido diseñada para su uso con el metaframework Roma pero también puede ser utilizada en otras soluciones que utilicen objetos java para definir la interfaz gráfica o aquellos frameworks que generen automáticamente la interfaz gráfica a partir del modelo como por ejemplo Ruby on Rails o Grails. Un mecanismo de plantillas como Janus adaptado a estos entornos permitiría la personalización de la interfaz por defecto proporcionada por estos frameworks, así como la integración de nuevas tenologías de presentación.

VIII. Conclusiones y Trabajos Futuros

En este artículo se ha presentado el generador de plantillas Janus, desarrollado para Romaframework, dentro del proyecto europeo Romulus, aun en fase de desarrollo.

El trabajo ha expuesto los diferentes motores de generación de plantillas existentes en la actualidad, centrándose en los dos más interesantes (Velocity y FreeMarker), comentando sus similitudes y diferencias. Posteriormente se ha definido la arquitectura del módulo Janus y se ha mostrado un caso de uso del mismo.

En el futuro se aplicará el módulo Janus a otros demostradores y se desarrollará una API Javascript para una mayor facilidad en el desarrollo. Además se implementarán varios estilos mediante plantillas que permitirán la creación de temas personalizados según el perfil del usuario.

Agradecimientos

El módulo Janus ha sido desarrollado para Romaframework, dentro del proyecto europeo Romulus ICT-2007.1.2, financiado por la Comunidad Europea dentro del marco Seventh Framework Programme y en fase de desarrollo desde enero de 2008.

Referencias

[1] Apache. The apache velocity project, 2009. Disponible en http://velocity.apache.org/.
[2] P. Bakaus and the jQuery UI Team. Jqueri ui project, 2009. Disponible en http://jqueryui.com/.
[3] Beilpuz. Herramienta beilpuz, 2009. Disponible en http://beilpuz.getmike.de/doku.php.
[4] S. Consortium. Scrooge, 2009. Disponible en http://scrooge.germinus.com.
[5] E. D. Corporation. Evans reports, official site, online. available:, 2007.
[6] A. creativity Solutions. Roma framework, the new way to conceive web applications, 2009. Disponible en http://www.romaframework.org/.
[7] eRuby. Herramienta eruby, 2009. Disponible en http://en.wikipedia.org/wiki/ERuby.
[8] GvTags. Herramienta gvtags, 2009. Disponible en http://www.gvtags.org/.
[9] h2o. Herramienta h2o, an elegant php template engine, 2009. Disponible en http://www.h2o-template.org/.
[10] D. Jesse. Yet another multicolumn layout — an (x)html/css framework, 2009. Disponible en http://www.yaml.de/en/.
[11] G. M. Luigi Dell'Aquila, Luca Garulli. Roma <meta> framework handbook, 2009. Disponible en http://www.romaframework.org/doc/RomaHandbook.pdf.
[12] T. Meeks. Comparing the google web toolkit to echo2, 2006. Disponible en http://www.theserverside.com/news/thread.tss?thread_id=40804.
[13] NextApp. Echo2 web framework, 2009. Disponible en http://echo.nextapp.com/site/echo2.
[14] F. project. Freemarker versus velocity, 2009. Disponible en http://freemarker.org/fmVsVel.html.
[15] F. project. Fremarker: Java template engine library - overview, 2009. Disponible en http://freemarker.org/.
[16] A. Protopsaltou. Model driven development with ruby on rails, it university of göteborg, 2004.
[17] Romulus. Annex i, description of work, 2007. From the Seventh Framework Programme Theme ICT-2007.1.2, p11.
[18] Romulus Consortium. Service and software architectures, infrastructures and engineering.romulus - domain driven design and mashup oriented development, 2009. Disponible en http://www.ict-romulus.eu/.
[19] D. P. Templates. Herramienta dwoo, 2009. Disponible en http://dwoo.org/.
[20] E. Templating. Herramienta evoque templating, 2009. Disponible en http://evoque.gizmojo.org/.
[21] I. The java Comunity Process (SM) Program: Alex Buckley, Sun Microsystems. Jsr 175: A metadata facility for the javatm programming language, 2009. Disponible en http://www.jcp.org/en/jsr/detail?id=175.
[22] J. van Bergen. Velocity or freemarker?, 2009. Disponible en http://www.javaworld.com/javaworld/jw-11-2007/jw-11-java-template-engines.html?page=3.
[23] Wikipedia. Entrada template engine (web), 2009. Disponible en http://en.wikipedia.org/wiki/Template_engine_(web).

Modelo de Pruebas de Software en el Desarrollo de Aplicaciones Orientadas a Servicios

Hugo A. Parada G., Juan C. Dueñas, Boni García
Departamento de Ingeniería de Sistemas Telemáticos,
Universidad Politécnia de Madrid
Avenida Complutense 30, Ciudad Univesitaria, 28040 Madrid, España.
{hparada,jcduenas,bgarcia}@dit.upm.es

***Resumen*- El modelo Arquitectura Orientada a Servicios (SOA) es una tecnología emergente para el desarrollo de software que afecta a la forma cómo se construyen las aplicaciones. Para aplicarlo es necesario adaptar las actividades del ciclo de vida de desarrollo a las exigencias planteadas por SOA. Aspectos metodológicos como procesos, procedimientos y técnicas deben cambiarse, para lograr su correcta aplicación. En este sentido la fase de pruebas es la más afectada, dado que el enfoque de pruebas a diferencia de las otras disciplinas de desarrollo se adapta más lentamente cuando surgen nuevos modelos. Por lo tanto para probar aplicaciones SOA, se tiende a aplicar las pruebas como en el modelo tradicional sin lograr buenos resultados. Como solución en este trabajo proponemos un modelo metodológico general de cómo aplicar las pruebas a las aplicaciones desarrolladas bajo SOA. Este modelo permitirá la definición de los procesos y de los artefactos y facilitará la ejecución de las pruebas a las aplicaciones SOA.**

Palabras Clave: Pruebas de software, modelo SOA, modelos de desarrollo-

I. INTRODUCCIÓN

La arquitectura orientada a servicios (SOA "Service Oriented Architecture") es una tecnología emergente para el desarrollo de software ampliamente aceptada por la industria [1]. Esta tiene como objetivo incrementar la eficiencia, la agilidad, y la productividad de las empresas. En este entorno la aplicación de las metodologías de desarrollo rápido y ágil es crucial para responder con rapidez a las necesidades de los usuarios. Sin embargo la calidad es un requisito clave que debe tenerse en cuenta para poder competir en el mercado. Adicionalmente el modelo SOA agrega más complejidad a este aspecto debido a que muchos de los servicios se desarrollan por terceros, de forma distribuida, a que varias de sus interacciones sólo se conocen durante la ejecución de la aplicación y que estas ocurren con servicios externos de los cuales solo se tiene acceso a su interfaz[2]. Por lo tanto es necesario usar métodos, técnicas y herramientas de pruebas apropiadas; para facilitar su aplicación y obtener los resultados esperados.

Este nuevo paradigma cambió la forma de construir las aplicaciones empresariales en cuanto a las actividades del ciclo de desarrollo [3][4]. Dentro de ellas las pruebas han sido una de las etapas más afectadas, debido a que no se han adaptado totalmente a las condiciones del modelo SOA.

En general el enfoque de pruebas a diferencia de las otras disciplinas de desarrollo se adapta más lentamente a las nuevas exigencias tanto técnicas como metodológicas cuando surgen nuevos modelos [5]. A pesar del avance y de la experiencia obtenida en los últimos años como resultado de la aplicación del modelo SOA, no existe un marco general que permita la aplicación de pruebas de manera estándar. En consecuencia se tiende a aplicar las pruebas de la misma manera que en las aplicaciones tradicionales (en cuanto al uso de métodos, técnicas y herramientas), con lo cual se añade mayor dificultad a su ejecución. Impactando de manera negativa la calidad, el tiempo y los costos de desarrollo [6].

Los dominios emergentes deben afrontar varios retos para probar las aplicaciones. Uno de los más críticos es la escasez de procesos y metodologías que guíen su aplicación. Como evidencia de ello en [6] se presentan los resultados de una revisión de la literatura sobre pruebas en el dominio SOA, entre los que se resalta que de las 127 referencias sobre pruebas SOA revisados solo dos (2) tratan sobre procesos de prueba. Igualmente se hace énfasis sobre lo importancia de definir un proceso de pruebas para este dominio y proponen algunos aspectos que debería incluir su definición entre los que menciona: procedimientos, productos, herramientas y otros recursos.

Las pruebas en sí mismas son una disciplina compleja que consume un elevado porcentaje de esfuerzo y recursos en relación al costo total del proyecto. Por ello en algunos casos se aplican de manera "informal", con escasa rigurosidad (en cuanto a que se dirigen por un proceso "intuitivo") y que se depende de la experiencia y habilidades del personal que las aplica. Al respecto un estudio realizado para el Instituto Nacional de Estándares y Tecnología de los Estados Unidos, cuantifica como alto el impacto económico de no tener una infraestructura adecuada de pruebas en las empresas; y entre los principales efectos señala: un incremento de los fallos, incremento en los costos de desarrollo de software, retraso en la puesta en funcionamiento del software en el mercado y un incremento en los costos en las transacciones comerciales [7]. Igualmente en este estudio se definen como componentes fundamentales de una infraestructura de pruebas: al conjunto de tecnologías, procesos, procedimientos y estándares para la prueba de software, las herramientas de soporte a las actividades de pruebas y las etapas que definen las actividades a realizar. Estos elementos conformarían un

entorno ideal y completo para la aplicación de pruebas. Esta es una necesidad en los dominios emergentes, y para facilitar su aplicación se adaptan las metodologías existentes [8].

En las pruebas de aplicaciones SOA, dependiendo del rol que esté involucrado existen diferentes perspectivas [9]: la del desarrollador del servicio, la del proveedor del servicio y el integrador del servicio. Todos tienen diferentes necesidades de pruebas. En este artículo proponemos un modelo general de pruebas que facilita la definición de los procedimientos técnicos y la aplicación de técnicas para probar los elementos básicos que componen una aplicación SOA basada en Java, en tecnologías HTML, XML y el lenguaje de descripción de servicios WSDL; en el entorno de desarrollo; es decir desde la perspectiva del desarrollador. Proporcionamos una guía para aplicar las pruebas del servicio en el entorno de desarrollo. Esta apoyará las actividades técnicas tanto para el desarrollador como para el equipo de pruebas. De la misma manera se definen las bases para facilitar el trabajo de los demás roles participantes en el desarrollo y despliegue de aplicaciones SOA. El trabajo continúa en el segundo capítulo con la presentación de algunos de los trabajos más relevantes en el área de desarrollo y de pruebas SOA. En el capítulo tres presentamos una vista general del modelo de pruebas propuesto y el enfoque usado para su organización metodológica. En el capítulo cuatro describimos el ciclo de pruebas y sus actividades. En el capítulo cinco detallamos el proceso de prueba para cada uno de los elementos básicos de una aplicación SOA en el entorno de desarrollo. En el capítulo seis discutimos las ventajas y las posibilidades de mejora del modelo propuesto y finalizamos con las conclusiones y las líneas trabajo futuro.

II. TRABAJOS RELACIONADOS

La disciplina de pruebas de software ha alcanzado la madurez. Esto se refleja en un marco teórico sólido, compuesto por estándares, técnicas, buenas prácticas y la evolución tecnológica que ha facilitado su aplicación. Sin embargo con la aparición de nuevos enfoques de desarrollo de software, esta debe actualizarse y para ello se proponen nuevas formas de aplicar sus postulados. En esta sección presentamos algunos de los trabajos que tratan la aplicación de pruebas a las aplicaciones SOA y que se relacionan con nuestra propuesta. Específicamente aquellos que proponen aspectos metodológicos como técnicas para diseñar, generar o ejecutar casos de pruebas.

En [10] se propone un modelo y un proceso de pruebas de componentes de software, para ser ejecutado por terceros, quienes reciben como entrada el componente a probar y los metadatos que expresan su funcionamiento junto con los requisitos que satisfacen. Como resultado se generan todas las actividades para probar el software (planificación, diseño, elaboración y ejecución del caso de prueba). Esta propuesta muestra un escenario que puede aplicarse para probar aplicaciones SOA, debido a la forma como se desarrollan (distribuidas y a que se usan servicios desarrollados por terceros). Sin embargo bajo el modelo propuesto, el control de los artefactos generados durante las pruebas y los procedimientos aplicados queda fuera del alcance del responsable del desarrollo de la aplicación.

Actualmente es común usar diferentes modelos de desarrollo (ágiles, distribuidos, por prototipos, etc.). Por lo tanto un proceso de pruebas debería poder aplicarse con independencia del modelo de desarrollo usado. En este sentido en [11] se proponen de manera genérica las características a tener en cuenta en la definición de un proceso de pruebas para adaptarlo a diferentes enfoques de desarrollo; como por ejemplo SOA. Allí se indican los elementos básicos que deberían tenerse en cuenta en la definición de un proceso, se define el concepto general de un proceso de pruebas, se sugieren las entradas, las salidas y las actividades concretas que deberían detallarse en una propuesta de proceso de pruebas, igualmente se caracterizan todos los elementos de proceso de un entorno de pruebas.

La construcción de aplicaciones SOA se basa en gran parte en la aplicación de las técnicas usadas en el desarrollo de Servicios Web. En consecuencia las técnicas de pruebas de Servicios Web y las herramientas también se aplican en este entorno. En [12] se basan en esta característica para proponer un procedimiento paso a paso para probar las conexiones que se establecen durante la orquestación de servicios bajo SOA y se define un procedimiento basado en el soporte que da la herramienta SoapUI. Este tipo de soluciones es fundamental cuando se aplica como parte de un proceso que soporta una metodología. Es en este sentido en el que dentro del modelo metodológico propuesto, se pueden incluir procedimientos similares para ampliar el alcance de las pruebas a aplicaciones SOA.

Respecto de la definición de modelos de prueba en [13] se propone un modelo específico de pruebas SOA dirigidas por modelos. En este trabajo se definen las etapas de diseño de pruebas, generación de casos de prueba y ejecución; para las que se definen las actividades necesarias para aplicar las pruebas. Esta aproximación a pesar de describir una técnica específica, propone unas fases que pueden tomarse como punto de partida para definir un proceso más general en cuanto a la aplicación de técnicas.

Un trabajo importante sobre la generación de casos de prueba funcional de servicios se expone en [14]. En él se define un procedimiento para extraer la información necesaria de la especificación de las operaciones de comportamiento (WSDL-S) y con ellas generar un conjunto de casos de prueba. Este trabajo se centra en la generación de casos de prueba; sin embargo existen otras fases que deben cubrirse como parte de un proceso de pruebas.

En la definición de una metodología de pruebas debe tenerse en cuenta que existen diferentes vistas en cuanto al interés de los participantes. De acuerdo con esta premisa en [9] se indican las perspectivas de acuerdo con los roles que participan en el modelo SOA y que técnicas se deben aplicar para satisfacerlos, de la misma forma se describen los niveles de aplicación de pruebas SOA. Al respecto en [15] se describen las dificultades de las pruebas a nivel unitario, de integración y funcional en las aplicaciones SOA. Igualmente se dan algunas indicaciones de cómo solucionarlas teniendo en cuenta la complejidad del modelo. Esto es útil para la construcción de un modelo metodológico de pruebas.

Un aspecto básico de la disciplina de pruebas es la aplicación de las técnicas tradicionales tales como las pruebas de caja blanca, caja negra y caja gris. En [16] se describe como aplicarlas a los Servicios Web sus ventajas y

desventajas. Sin embargo no se expresan los detalles técnicos ni los procedimientos para su aplicación.

Como se muestra en el análisis de los trabajos relacionados con las pruebas bajo el modelo SOA, estas se centran en ofrecer soluciones a situaciones puntuales tales como la generación de casos de pruebas de manera automática, los niveles en los que deberían aplicarse las pruebas, las perspectivas a considerar durante la ejecución de las pruebas o la definición de procedimientos basados en herramientas. Sin embargo no se propone un proceso general que permita aplicar las pruebas que incluya procedimientos para sus fases más importantes como la planificación, la especificación, la generación y la ejecución de los casos de prueba.

III. MODELO DE PRUEBAS PROPUESTO

La forma más común de implementación del modelo SOA se basa en el desarrollo de Servicios Web [17]. Estos implementan la lógica de una unidad de negocio de manera independiente. Como se muestra en la figura 1 (que representa el esquema de la aplicación usada en el caso de estudio), la implementación de la lógica del servicio se puede basarse en: clases java (POJO – Plain Old Java Object), elementos de aplicaciones web (páginas HTML, ficheros XML, beans que describen clases, código JavaScript, objetos de acceso a datos –DAO data acces object-, etc). Un servicio puede invocar a otros servicios. Adicionalmente para tener acceso al servicio se define una interface por medio de un lenguaje (generalmente se usa WSDL – Web Service Description Language). Por lo tanto para garantizar la calidad de los servicios desarrollados, el desarrollador de servicios debería probar todos los componentes que lo conforman. Así la perspectiva del desarrollador presenta un panorama complejo. Es decir el sistema bajo prueba (SUT) puede contener varios elementos (ver fig. 1) y para probarlos es necesaria la definición de un proceso y de técnicas soportadas por herramientas.

Fig. 1. Elementos del sistema bajo prueba usado en el caso de estudio

Nuestra propuesta tiene como objeto proponer un modelo de pruebas que facilite al desarrollador la aplicación de las pruebas. Para ello partimos de la descripción de un ciclo de pruebas. Este ciclo se compone de fases que a su vez se dividen en tareas o actividades y procedimientos.

Las fases del ciclo de pruebas definen de manera general las actividades que deben realizarse. Se describen aspectos como los artefactos de entrada y de salida de cada fase. Igualmente se describen aquellos que no son parte del entorno de pruebas pero que son necesarios para ejecutar el proceso (por ejemplo los requisitos del sistema).

La definición del ciclo de pruebas y sus actividades son básicas para describir el proceso, no obstante no se aplican al entorno del desarrollador. Sin embargo permiten instanciar las actividades técnicas que guiarán su aplicación por el desarrollador (probador) en el entorno de desarrollo.

La aplicación de este proceso se basa en una estrategia ascendente. Sin embargo para definirlo hemos seguido un enfoque descendente (ver fig. 2), que facilita la instanciación del proceso y las actividades a partir de una base metodológica. Este enfoque se divide en cuatro niveles así: de gestión de procesos, de gestión de proyectos, de ejecución del proyecto y de ejecución técnica.

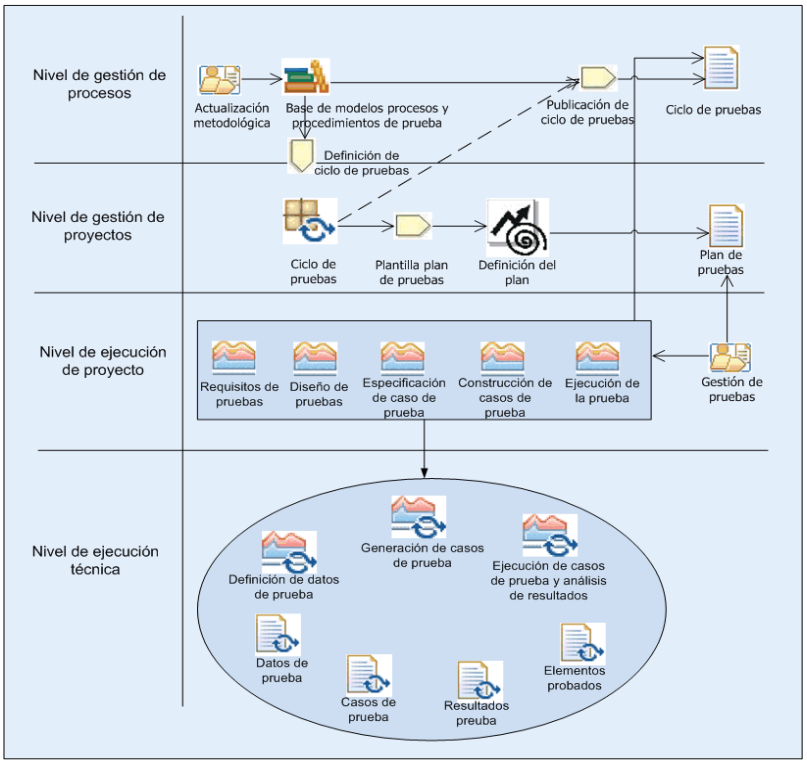


Fig. 2. Modelo de instanciación del proceso de pruebas.

En el nivel de gestión de procesos a partir de una base metodológica de pruebas definimos un ciclo de pruebas que corresponde al nivel de gestión de proyecto de pruebas. A partir de este se definen un plan de proyecto y se crean las actividades y procedimientos técnicos que conforman el nivel de ejecución del proyecto. Para la definición de las tareas del nivel de ejecución técnico se tienen en cuenta aquellas actividades que implementan procedimientos relacionados con la generación de los casos de prueba, su ejecución y el informe de resultados.

Las actividades técnicas (que se muestran en el nivel de ejecución técnica) definen paso a paso el procedimiento a seguir para probar los elementos del sistema bajo prueba (SUT). Para la perspectiva del desarrollador hemos definido actividades para las pruebas unitarias de POJOs, de elementos de aplicaciones web y unitarias de servicio. Existe mucha similitud entre las actividades a realizar sin embargo varía la aplicación de las herramientas tanto de generación de casos de prueba como de ejecución.

IV. Ciclo de pruebas

Como lo mencionamos previamente la definición del ciclo de pruebas es básica para definir las interacciones con las fases del proceso de desarrollo y para describir las actividades técnicas de pruebas. El ciclo de aplicación de pruebas propuesto se divide en cinco fases [18]: definición de requisitos de prueba, diseño de casos de prueba, especificación de la prueba, implementación de la prueba y ejecución de los casos de prueba. Como se muestra en la figura 2 estas actividades permiten ejecutar las pruebas como un proyecto.

A. Definición de requisitos de prueba

Esta fase tiene como objetivo configurar y preparar el entorno de pruebas. Se divide en cinco actividades: configuración del entorno de pruebas, selección y/o actualización de las herramientas de prueba, definición del sistema bajo prueba, elaboración del plan de pruebas. Esta fase tiene como artefactos de entrada el código, la documentación y los requisitos del sistema. Como salida genera las características del sistema a probar (tamaño, componentes), herramientas de prueba instaladas y configuradas, listado de variables del sistema modificadas y el plan de pruebas.

B. Diseño de los casos de prueba

Es una actividad técnica que tiene como objetivo aplicar la estrategia elegida para la prueba. Para realizar esta actividad es necesario como mínimo tener acceso al código de los elementos a probar, a los requisitos que satisface y al plan de pruebas. Está compuesta por dos actividades: la definición de las técnicas y elementos a probar y la definición de los casos de prueba y los requisitos a probar. Como resultado se generan: los elementos bajo prueba junto con las técnicas de prueba, listado de casos de prueba y criterios de prueba.

C. Especificación de la prueba

Tiene como objetivo definir las plantillas de los casos de prueba y el orden en que deben ejecutarse los casos de prueba. Requiere como entrada la definición de los casos de prueba y las herramientas de prueba. La definición del orden de ejecución describe la estrategia de ejecución de las pruebas. Básicamente se indica si el caso se ejecutará de manera aislada o como parte de una suite y en qué orden debería ejecutarse, sí fuese necesario ejecutar varios casos. También se define que elementos del SUT deben lanzarse e inicializarse para poder realizar la prueba.

D. Implementación del caso de prueba

En la fase de implementación de la prueba se genera el código de los casos de prueba y se aplican las estrategias de generación de casos de prueba (automática, grabación de la interacción con el sistema, manual, combinada, etc.). La generación de los casos de prueba se basa en plantillas para mejorar la eficiencia en la obtención de los casos. El resultado de esta fase es el código de los casos de prueba (que debería estar documentado).

E. Ejecución de la prueba

En esta fase se ejercita el sistema o elemento bajo prueba. Para ello es necesario verificar nuevamente las condiciones del contexto de la prueba y la configuración de la plataforma necesaria para el despliegue de la prueba. Los resultados generados se analizan para tomar las decisiones respecto al código. Si pasa la prueba se podrá avanzar en la siguiente iteración o liberación de una nueva versión. En caso contrario se genera una solicitud de corrección del fallo.

V. Pruebas de elementos SOA en el entorno de desarrollo

Para aplicar las pruebas a los elementos SOA, es necesario instanciar las actividades definidas en el ciclo de pruebas. Para ello definimos para cada elemento los artefactos que participan, las herramientas que soportan el proceso y se describe específicamente el ciclo de prueba. Es decir la interacción con las herramientas propias del entorno y los resultados esperados en cada caso.

El proceso definido cubre las pruebas unitarias de POJOs elementos de aplicaciones web y de servicios en el entorno de desarrollo. Para su aplicación proponemos una estrategia ascendente. Se parte de las pruebas unitarias de clases hasta incluir las pruebas del servicio. Por lo tanto al probar el servicio es probable que algunos de los elementos que lo conforman ya hayan sido probados. Este tipo de pruebas lo llamamos pruebas agregadas. Para representar gráficamente el proceso usamos los símbolos definidos por el estándar SPEM [19], así proporcionamos un lenguaje común que facilita la comunicación y la coordinación en el equipo del proyecto. En los gráficos de procesos detallados los enlaces se etiquetan con *<<in Mand>>* para indicar que es una entrada obligatoria para el proceso; y *<<Out Mand>>* para señalar las salidas del proceso.

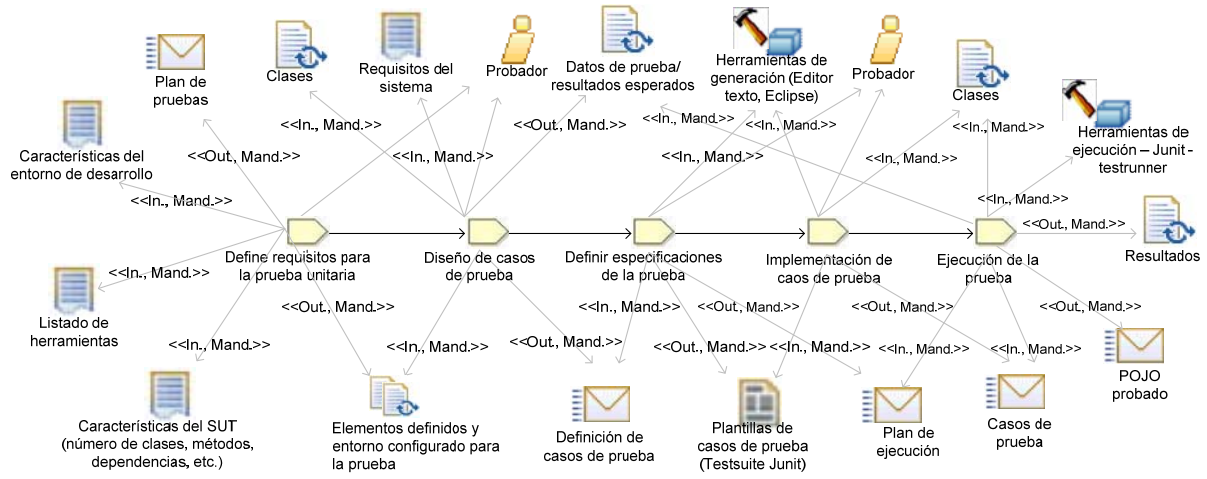


Fig. 3. Proceso de pruebas unitarias de POJOs

A. *Pruebas unitarias de POJOs*

En la figura 3 se muestran las principales actividades que deben cubrirse para realizar la prueba unitaria de un POJO. Para cada una de las etapas del ciclo de vida se definen las entradas necesarias así como el resultado de cada una de ellas. Todas las actividades del proceso están relacionadas. A partir de la fase de definición de requisitos (definición de los elementos necesarios para poder ejecutar la prueba) cada fase usa las salidas generadas por sus predecesoras como entrada para poder completar una iteración del ciclo de pruebas unitarias para un POJO.

El proceso inicia con la fase de definición de los elementos o requisitos necesarios para las pruebas. Toma como entradas: el listado de herramientas de prueba, características del entorno de desarrollo y características de la clase. Con ello genera como salida el entorno de pruebas configurado y los elementos definidos para poder realizar las pruebas.

La etapa de diseño de los casos de prueba toma como entradas el entorno configurado, el código de la clase y los requisitos de la clase. Produce como salida la definición de los casos de prueba y qué métodos de las clases bajo prueba se ejercitarán.

En la definición de especificaciones de la prueba se definen las plantillas de los casos de prueba (casos JUnit), la forma como se ejecutarán, en este caso desde las clases más sencillas a las más complejas y con dependencias de otras. También se genera como resultado el plan de ejecución de los casos de prueba. El cual define el orden de ejecución de los casos en la suite de pruebas.

Durante la fase de implementación de los casos de prueba se escribe el código de la prueba. Tiene como entrada el código de la clase o POJO a probar, un editor y una plantilla de código de prueba si existiese. Como resultado se obtiene el código de la prueba.

Para la fase de ejecución de la prueba se requieren como entradas: el código de la clase, los datos de prueba, JUnit y el plan de ejecución, que generalmente es un script Ant que indica el orden de ejecución de los casos de prueba dentro de la suite de pruebas. Como resultado se obtienen los informes con el resultado de la prueba y el POJO probado.

B. *Pruebas de elementos de aplicaciones web*

En la figura 4 se muestra el proceso de pruebas unitarias de los elementos de aplicaciones web. Básicamente el proceso se conforma de los mismos pasos del proceso de pruebas de un POJO. Aunque se generan los mismos artefactos en su implementación (que es la que describe la figura) se presentan algunas diferencias principalmente debido a los siguientes aspectos:

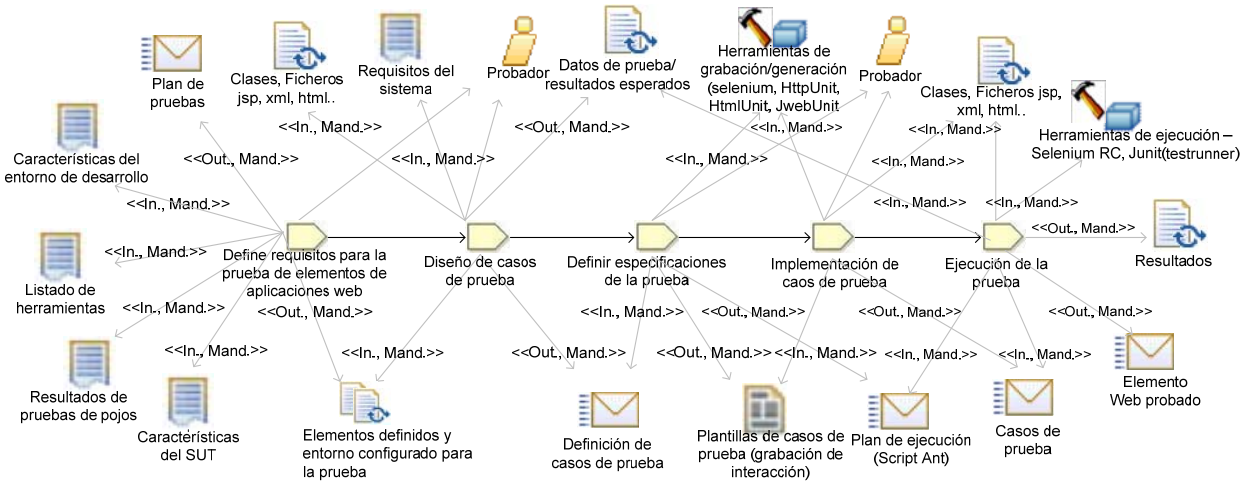


Fig. 4. Proceso de pruebas de elementos de aplicaciones web

- Gran parte de la lógica de las aplicaciones web se desarrolla por los POJOs a los que ya se les han aplicado las pruebas unitarias (se requiere como entrada *los resultados de prueba de POJOS*),
- ya se ha cumplido al menos una iteración completa del proceso de desarrollo,
- para poder ejecutar las pruebas de los elementos de aplicaciones web es necesario que exista una versión de la aplicación que permita ejercitar una funcionalidad completa. Es decir varios componentes integrados que satisfacen un grupo de requisitos. Esto se puede validar aplicando una prueba con menor rigor que solo tiene como objetivo evaluar si el código es ejecutable o no.

Para que el proceso pueda ejecutarse cada una de las actividades necesita de entradas que posibilitarán avanzar hacia la siguiente actividad. Sin embargo podemos considerar como la entrada principal del proceso los elementos de la aplicación web integrados, los resultados de las pruebas de los POJOs y las herramientas para desarrollar las pruebas específicamente: HtmlUnit, HttpUnit, Selenium y JWebUnit.

Las herramientas usadas para soportar las actividades específicas del proceso (generación y ejecución de casos de prueba) proporcionan ventajas en cuanto al esfuerzo dedicado para ejecutar nuevos casos (ejecución de varias iteraciones). Para ello es necesaria la preparación de algunos artefactos como los datos de prueba y las plantillas (que para este caso contiene la grabación de la interacción del usuario con los elementos del SUT). Con estos dos artefactos podremos ejecutar un gran número de casos de prueba adicionando más datos de prueba al artefacto durante la fase de diseño de casos de prueba. Así el esfuerzo inicial invertido en la generación del caso de prueba de acuerdo con el proceso se compensa con la reutilización del caso y la cobertura obtenida por su aplicación con diferentes datos.

C. *Pruebas unitarias de servicios*

El proceso de pruebas de servicio que se muestra en la figura 5 describe los pasos que deben seguirse para probar el servicio de manera aislada; es decir una prueba unitaria de servicio. Dado que aplicamos el ciclo de vida de pruebas definido anteriormente, el proceso que proponemos cumple con las mismas fases que los procesos de pruebas unitarias de POJOs y de elementos de aplicaciones web, algunas de sus fases son exactamente iguales, pero otras varían de manera significativa.

La etapa de de definición de requisitos es básicamente la misma en los tres procesos, esta consiste en configurar y establecer el entorno de pruebas a partir del contexto de desarrollo. Luego en un entorno de desarrollo local esta fase, para la prueba de servicios demandaría muy poco esfuerzo.

La etapa de diseño de casos de prueba de servicios es exactamente igual que las descritas en los anteriores procesos. Tiene como objetivo definir los datos de prueba y los casos de prueba suficientes.

Las etapas de definición de especificaciones, de implementación y ejecución, son diferentes a las descritas anteriormente, debido a que su aplicación depende de la herramienta de prueba que se usa. En este caso usamos SoapUI que permite definir el caso de prueba a partir de la definición de la interface del servicio. Los casos se agrupan en suites de pruebas. Para su ejecución cada caso de prueba definido por SoapUI se divide en pasos. De esta forma se puede controlar el proceso de prueba.

De la misma manera que en las pruebas de aplicaciones Web podemos ejecutar el caso de pruebas de forma iterativa sin cambiar el código del caso de prueba. Para ello podemos usar scripts Groovy o la herramienta PushToTest para inyectar datos al caso de prueba durante su ejecución, así podremos probar el servicio tantas veces como datos tenga el artefacto datos de prueba (fichero en formato csv).

VI. DISCUSIÓN

El modelo de pruebas propuesto define con un gran nivel de detalle todas las actividades necesarias para aplicar pruebas de software en un entorno de desarrollo. La definición de un ciclo de vida permite organizar las actividades en etapas funcionales. Esto facilita su aplicación a diferentes modelos de desarrollo independientemente de si es ágil, tradicional, distribuido etc. Por lo tanto su adopción implicaría un menor esfuerzo.

Es común que en los entornos de desarrollo el resultado de las pruebas que se aplican en cuanto a eficiencia y eficacia dependen en gran parte de las habilidades del desarrollador-probador. Debido a que no dispone de una documentación formal que indique que aspectos debería tener en cuenta para aplicar una prueba y que procedimientos seguir.

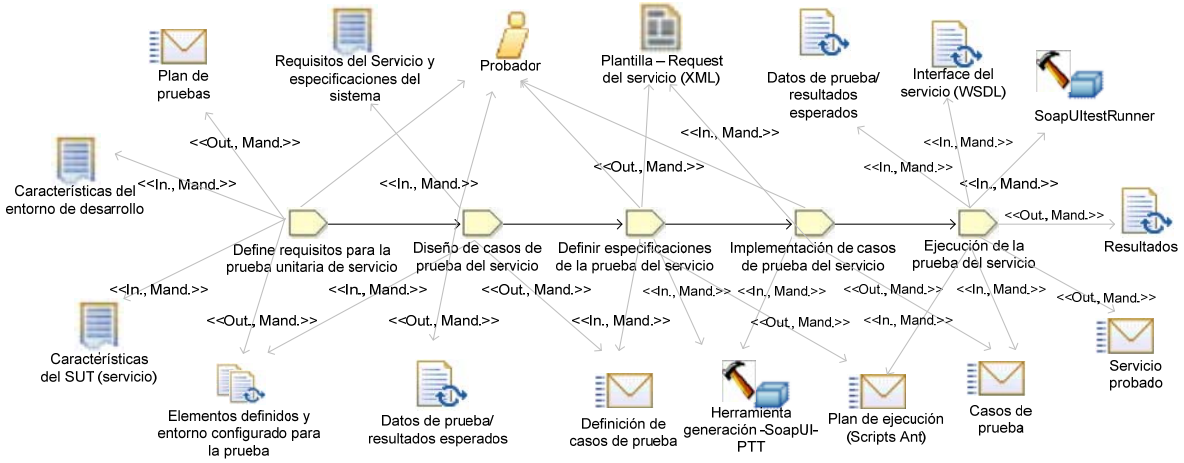


Fig. 5. Proceso de prueba unitaria de servicios

Este aspecto es más crítico en disciplinas emergentes como SOA, en las cuales se tienen que adaptar las técnicas y las herramientas de pruebas. En este sentido nuestra propuesta soporta conceptualmente la definición de los procedimientos de prueba en dos niveles: uno general y otro específico. A nivel general se contribuye con la formalización de aspectos metodológicos como las fases, los artefactos que sirven como entradas y los que se generan como salida. Esto facilita su extensión a dominios de aplicación específica. A nivel específico nos basamos en las herramientas que soportan las actividades de prueba para definir los procedimientos técnicos que guían las pruebas.

En el caso del modelo SOA la combinación del procedimiento tanto para prueba de servicios con la herramienta SoapUI como para los elementos de aplicaciones web con Selenium con los procesos instanciados del ciclo de vida, proporcionan un entorno eficiente para aplicar las pruebas. Dado que en la generación de los casos de prueba se logra automatizar la mayor parte del proceso, el desarrollador solo tiene que introducir pequeños cambios al código de la prueba. En cuanto a la ejecución de los casos de prueba se proporcionan mecanismos de inyección de datos usando el mismo caso de prueba. Con lo cual se garantiza un mayor nivel de cobertura con un menor esfuerzo.

Otra ventaja que se obtiene de la definición detalla de las fases del proceso es la integración de las tareas mediante scripts (Ant) que reciben los detalles del proceso como propiedades y que configuran el entorno de pruebas reduciendo al mínimo el esfuerzo durante la ejecución de las pruebas.

VII. Conclusiones y trabajo futuro

El modelo de prueba propuesto contribuye con la definición de un marco de pruebas para las aplicaciones SOA. Se define un proceso general que puede ser aplicado a cualquier modelo de desarrollo. En cuanto a las pruebas de aplicaciones SOA inicialmente se satisface la perspectiva del desarrollador del servicio. Sin embargo se sientan las bases para su extensión a las otras perspectivas (la del integrador y la del proveedor de servicios).

Por otra parte mediante el uso de herramientas Open Source hemos logrado automatizar algunas de las fases del proceso. Adicionalmente nos hemos apoyado en herramientas como Ant y Groovy para integrar por medio de scripts la ejecución de pruebas de los POJOs, elementos de aplicaciones web y de servicios.

Los resultados de este trabajo han sido aplicados en el marco del proyecto ITECBAN, en el cual hemos definido una metodología para instanciar las actividades técnicas y hemos creado los procedimientos para aplicar las herramientas de pruebas como parte integral del proceso. Continuamos investigando para definir un proceso de pruebas de integración y funcionales de servicios soportados por sistemas de integración continua.

Agradecimientos

El trabajo que se presenta ha sido realizado en el marco del proyecto ITECBAN: Metodología e infraestructura para el Core Bancario, financiado por el CDTI, bajo subcontratación de la empresa INDRA.

Referencias

[1] Ferguson, D. F., Stockton, M. L., ‘‘Service-oriented Architecture: Programming Model and Product Architecture,’’ *IBM Systems Journal*, 44, No. 4, pp. 753–780, 2005.

[2] Dave, S., Best Practices for Building SOA Applications, *SOAWORDL Magazine*, Sept. 21 2006. Online: http://soa.sys-con.com/node/275111, accessed: 2009.

[3] Tsai, W.T.; Gao, J.; Xiao Wei; Yinong Chen, "Testability of Software in Service-Oriented Architecture," *Computer Software and Applications Conference, 2006. COMPSAC '06. 30th Annual International*, vol.2, no., pp.163-170, 17-21 Sept. 2006.

[4] Tsai, W.T.; Jin, Z.; Wang, P.; Wu, B., "Requirement Engineering in Service-Oriented System Engineering," *e-Business Engineering, 2007. ICEBE 2007. IEEE International Conference on*, pp.661-668, 2007.

[5] Yoon, H., Ji, E., and Choi, B. 2008. Building test steps for SOA service orchestration in web service testing tools. In *Proceedings of the 2nd international Conference on Ubiquitous information Management and Communication, ICUIMC '08*, ACM, New York, NY, pp. 555-557, 2008.

[6] Farooq, A., Georgieva, K., and Dumke, R. R., Challenges in Evaluating SOA Test Processes. In *Proceedings of the international Conferences on Software Process and Product Measurement*, R. R. Dumke, R. Braungarten, G. Büren, A. Abran, and J. J. Cuadrado-Gallego, (Eds.), Lecture Notes In Computer Science, vol. 5338, Springer-Verlag, Berlin, Heidelberg, pp. 107-113, 2008.

[7] NIST. The economic impacts of inadequate infrastructure for software testing, May 2002. In line: http://www.nist.gov/director/prog-ofc/report02-3.pdf, (accessed Feb. 2009).

[8] Bertolino, A., "Software Testing Research: Achievements, Challenges, Dreams," *Future of Software Engineering, 2007. FOSE '07*, ISBN: 0-7695-2829-5, pp.85-103, 23-25 May 2007.

[9] Canfora, G.; Di Penta, M., "Testing services and service-centric systems: challenges and opportunities," *IT Professional* , vol.8, no.2, pp.10-17, March-April 2006.

[10] Jing G., Yuqing L., Maozhong J., "A Model of Third-Party Integration Testing Process for Foundation Software Platform," *Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for* , vol., no., pp.1199-1204, 18-21 Nov. 2008.

[11] Londesbrough, I., "A Test Process for all Lifecycles," *Software Testing Verification and Validation Workshop, 2008. ICSTW '08. IEEE International Conference on* , vol., no., pp.327-331, 9-11 April 2008.

[12] Yoon, H., Ji, E., Choi, B., Building test steps for SOA service orchestration in web service testing tools. In *Proceedings of the 2nd international Conference on Ubiquitous information Management and Communication* (Suwon, Korea, January 31 - February 01, 2008). ICUIMC '08. ACM, New York, NY, pp. 555-557, 2008.

[13] Lenz, Ch., Chimiak-Opoka J., Breu R., Model driven testing of soa-based software. In Daniel Luebke, editor, *Proceedings of the SEMSOA Workshop 2007 on Software Engineering Methods for Service-Oriented Architecture*, volume 244 of *CEUR Workshop Proceedings (ISSN 1613-0073)*, Hannover, Germany, pp. 99-110, May 2007.

[14] Sinha, A., Paradkar, A., Model-based functional conformance testing of web services operating on persistent data. In *Proceedings of the 2006 Workshop on Testing, Analysis, and Verification of Web Services and Applications* (Portland, Maine, July 17 - 17, 2006). TAV-WEB '06. ACM, New York, NY, pp. 17-22, 2006.

[15] Ribarov, L., Manova, I., Ilieva, S., Testing in Service Oriented World. In *Proceedings of the International Conference on Information Technologies* (InfoTech-2007), vol. 1, September 21-23, Bulgaria, 2007

[16] Mallal, R., Yunus, M., SOA Testing using Black, White and Gray Box Techniques. Online : http://www.softwaremag.com/l.cfm?doc=958-6/2006 , accessed 2009.

[17] Erl, T., Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services, Prentice Hall PTR, ISBN: 0-13-142898-5, April 2004.

[18] Baker, P., Ru Dai, Z., Grabowski, J., Haugen, Ø., Schieferdecker, I., Williams, C., Model Driven Testing – Using the UML Testing Profile, Springer, 2008.

[19] OMG, “Software Process Engineering Metamodel Specification”, version 2.0, Final Adopted Specification, ptc/07-03-03, on line: http://www.omg.org/docs/ptc/07-03-03.pdf, accessed 2009.

Análisis Macroscópico de los Dominios .es

Manuel Álvarez, Fidel Cacheda, Alberto Pan
Departamento de Tecnologías de la Información y las Comunicaciones
Universidade da Coruña (Spain)
Facultade de Informática, Campus de Elviña s/n, 15071
{mad, fidel, apan}@udc.es

***Resumen*- El objetivo final de este estudio consiste en la caracterización de los sitios web de los dominios .es para determinar el nivel de utilización de tecnologías de Web Oculta de lado cliente. Para la caracterización se utiliza la página principal de los servidores web de los dominios .es. El estudio tiene lugar en dos fases. En la primera fase se realiza un *crawling* de profundidad 1 sobre todos los dominios .es para determinar cuántos están activos, con servidor HTTP habilitado, y cuántos de ellos presentan redirecciones HTTP a otros servidores. En la segunda fase (que queda fuera del alcance de este artículo) se realizará un análisis del contenido de cada una de las páginas obtenidas en la primera fase, para determinar qué tecnologías utilizan y caracterizarlos en función de la complejidad requerida por un *crawler* que quiera incluirla entre sus sitios de recopilación de información.**

***Palabras Clave*- Crawling, Web Oculta, DNS, Ajax.**

I. INTRODUCCIÓN

La aproximación utilizada más habitualmente para recopilar y localizar información en Internet la constituyen los buscadores basados en técnicas de *crawling*. Los *crawlers* son programas software capaces de recorrer la Web automáticamente, recopilando las páginas accedidas para construir un índice que permita búsquedas sobre su contenido. Sin embargo, los *crawlers* actuales, sólo pueden acceder a la parte de la Web que se encuentra publicada y enlazada como páginas estáticas. Aunque estas páginas representan una gran cantidad de información, constituyen sólo una pequeña porción de toda la información web disponible. Existe gran cantidad de información que es generada dinámicamente por un servidor en respuesta a acciones del usuario. A esta porción de la Web suele denominársele 'Web Oculta' (*Hidden Web*) o 'Web Profunda' (*Deep Web*) [1].

Desde el punto de vista de las técnicas de *crawling*, el problema de tratar la Web Oculta puede dividirse en dos retos principales:

- *Crawling* del 'lado servidor'. Gran cantidad de sitios web ofrecen formularios de búsqueda para acceder a recursos almacenados en bases de datos subyacentes. Esta información no es accesible desde *crawlers* convencionales, porque estos no tienen la capacidad para ejecutar consultas sobre los formularios.
- *Crawling* del 'lado cliente'. Muchos sitios web usan tecnologías tales como lenguajes de script del lado cliente (e.g. JavaScript), y/o complejos sistemas de mantenimiento de sesión. Los *crawlers* convencionales no pueden alcanzar este tipo de páginas, puesto que no son capaces de ejecutar los scripts y no pueden mantener o recrear las sesiones adecuadas.

La Web Oculta del lado cliente ha cobrado mayor importancia en los últimos tiempos debido al uso de las tecnologías de interactividad en el cliente web basadas en *Ajax* [2]. El nivel de utilización de lenguajes de script en el diseño de sitios web ha ido variando a lo largo del tiempo. Tras el gran auge de sus comienzos, fue perdiendo protagonismo, en gran parte debido a las dificultades de los *crawlers* globales para acceder a su información. Actualmente, y principalmente desde la aparición de tecnologías como Ajax y a un nuevo modelo de diseño de sitios web en los que el lado cliente gana mucha importancia, las tecnologías de *scripting* han resurgido con fuerza.

Desde el punto de vista de los sistemas de *crawling* global sería útil, no sólo conocer el porcentaje de fuentes en Internet que utilizan este tipo de tecnologías en la actualidad, sino también para qué las están utilizando. De esta forma sería posible determinar si están justificados los esfuerzos orientados a la construcción de sistemas de *crawling* que permitan acceder a la información contenida en la Web Oculta del lado cliente.

El presente estudio se centra en los sitios web de los dominios españoles. Para ello, se ha planteado un *crawling* de nivel de profundidad 1, mediante el que obtener y analizar la página principal de cada uno de los servidores web de los dominios .es.

En una primera fase del estudio se pretende realizar una caracterización de los dominios .es en función de los siguientes parámetros:

- número de nombres de dominio que no están dados de alta en servidores de DNS,
- número de nombres de dominio que están asociados a un servicio de *parking*, es decir, que están dados de alta en un servidor de DNS pero no poseen un servidor propio, sino que han contratado un servicio externo que genera de forma automática una página con anuncios para ese dominio,
- número de nombres de dominio que tienen servidor web,
- número de nombres de dominio que utilicen HTTP *redirects* para redirigir a otros.

En una segunda fase se realizará un análisis del contenido de cada una de las páginas obtenidas en la primera fase, para

determinar qué tecnologías utilizan y caracterizarlas en función de la complejidad requerida por un *crawler* que quiera incluirlas como sitios de inicio. En esta segunda fase se trata de definir una serie de niveles de utilización de tecnologías de *script*, estimando para cada uno de ellos el coste que supondría a un sistema de *crawling* tratarlo. La escala podría tener en cuenta aspectos tales como: sitios que requieren usuario/contraseña, enlaces estáticos, enlaces con *script* en el atributo `href`, enlaces con *script* en algún manejador de evento, generación de código HTML utilizando la función `document.write`, etc.

La escala definida se utilizará para dos propósitos: por una parte, para determinar el nivel de utilización de las tecnologías del lado cliente en la Web de hoy y, por otra parte, para caracterizar el nivel de efectividad de diferentes sistemas de *crawling* en el tratamiento de las tecnologías del lado cliente.

Este artículo describe los resultados preliminares de la primera fase del estudio.

Los siguientes apartados describen la arquitectura del sistema de *crawling* utilizado para obtener la información de dominios, analizarlos y consultarlos, los resultados obtenidos, conclusiones y trabajos futuros.

II. ARQUITECTURA

Un sistema de *crawling* global convencional está compuesto básicamente por una lista de URLs de inicio de *crawling*, un cliente HTTP, un analizador de páginas para obtener enlaces a nuevos documentos a partir de la página procesada y un validador de URLs para eliminar aquellas que ya han sido previamente accedidas o superan un nivel de profundidad determinado, configurado antes de comenzar la ejecución del proceso de *crawling*.

La tarea de *crawling* propuesta difiere bastante de una tarea de *crawling* convencional, debido a que la lista de URLs iniciales contiene todos los recursos a obtener, y no se generarán nuevos URLs a recursos como resultado de analizar las páginas recolectadas.

En la Fig. 1 se muestra la arquitectura del sistema definido. En la parte superior aparece el módulo de *crawling*, que se alimenta de la lista de dominios a los que acceder, para obtener el estado actual para cada uno de ellos (en términos de resolución DNS y servicios HTTP). En la figura también se muestra el módulo responsable del análisis de los dominios recolectados, para la generación de diferentes estadísticas. Por último, también se ha desarrollado una herramienta web que permite realizar consultas sobre la información almacenada para cada dominio o las estadísticas generadas.

El sistema se ha implementado en el lenguaje de programación Java, utilizando las tecnologías Spring e Hibernate para la implementación de la capa modelo y Apache Tapestry para la capa web.

La información de entrada y salida del sistema se obtiene y vuelca a diferentes tablas de una base de datos relacional. Adicionalmente, se han creado diferentes utilidades de línea de comandos para precargar las tablas de datos de entrada a partir de documentos CSV y para iniciar los procesos de *crawling* y posterior análisis de resultados.

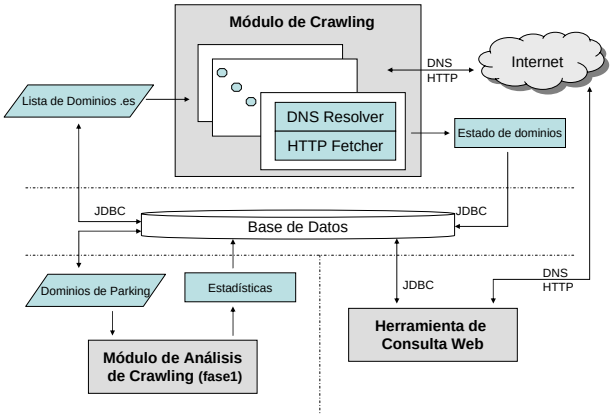


Fig. 1. Arquitectura del Sistema de Crawling

A. *Módulo de Crawling*

El módulo de *crawling* consta de un pool de componentes que utilizan dos conocidas librerías Java para realizar la resolución de nombres de dominios y la obtención de las páginas de inicio de los servidores HTTP.

El *DNS resolver* utiliza la librería opensource DnsJava [3] y como cliente HTTP se utiliza la librería opensource Jarkarta Commons HttpClient [4].

La entrada del proceso de *crawling* está formada por el conjunto de nombres de dominios de la jerarquía .es. La salida está formada por la información obtenida del servidor de DNS para el dominio considerado, y diversa información obtenida del servidor web a la hora de recuperar el recurso solicitado (la página de inicio del sitio web).

En los siguientes apartados se describe en detalle el funcionamiento de cada uno de estos componentes: *DNS resolver* y *HTTP fetcher*.

Es importante destacar que para minimizar el tiempo empleado en la tarea de *crawling* definida, es necesario disponer de un servidor de DNS que soporte un gran número de peticiones simultáneas, pues constituye el cuello de botella para el sistema (se accede a gran cantidad de dominios, pero el número de peticiones por servidor web va a ser muy reducido).

1. *Resolución DNS*

Aunque la resolución DNS es algo que los clientes HTTP realizan de forma transparente, el sistema propuesto debe de considerar un componente especial por dos razones:

- Para poder determinar si el nombre de dominio está dado de alta en el servicio de nombres, para permitir diferenciar el caso de un dominio sin DNS de aquel otro que sí está dado de alta en el servicio de DNS pero no presenta servidor web.
- Para obtener el nombre del servidor DNS autoritativo para el dominio considerado.

La obtención del servidor DNS autoritativo tiene lugar realizando la petición de un registro DNS de tipo SOA (*Start Of Authority*) para el dominio consultado. El registro de inicio de autoridad es el primer registro de recursos en cualquier archivo de zona de sistema de nombres de dominio (DNS), e indica el nombre del servidor DNS que constituye

la mejor fuente de información de los datos dentro de ese dominio DNS. Para obtener el valor del registro SOA es necesario realizar una consulta DNS recursiva, lo cual involucra una mayor carga de los sistemas DNS consultados, al no admitir como válida la respuesta de un servidor que no sea el autoritativo para el dominio.

La obtención del servidor de DNS autoritativo para un nombre de dominio es necesaria para determinar si el dominio está en *parking*. El *parking* de dominios permite tener de forma automática una página en un dominio llena de anuncios, por los que se obtienen unos beneficios por cada clic que un usuario realiza sobre ellos. Normalmente es interesante para aquellos que tienen dominios de palabras genéricas que atraigan mucho tráfico y no quieran realizar una inversión en desarrollarlos (como pornosporinternet.es que apunta a un *parking* de dominios). Estos dominios pueden generar decenas de miles de euros al mes sólo por el volumen de visitas *type-in* que reciben, sin aparecer en buscadores. Las visitas *type-in* son las de los usuarios que teclean la dirección a la que quieren ir en la barra del navegador. Representan entre un 10% y un 15% de las visitas de todo Internet (esto incluye la gente que teclea google.es en la barra del navegador, o yahoo.es).

A partir de una lista de servidores de *parking* de dominios se determinará si un nombre de dominio está en *parking* en función de si su servidor de DNS autoritativo se encuentra en esa lista.

Para finalizar, comentar que se ha configurado el tiempo máximo de espera de respuesta del servidor DNS a 10 segundos, y se ha configurado el sistema para consultar un servidor de DNS que escale a la demanda de peticiones a la que va a ser sometido (servidores de DNS de Telefónica).

2. *Cliente HTTP*

Para aquellos nombres de dominio para los que se haya determinado que se encuentra dado de alta en un servidor de nombres, es necesario comprobar si tiene servidor web activo y obtener su página de inicio.

Para la realización de la petición HTTP se utiliza un `User-Agent` especial, y diferente al de otros robots o navegadores: `Danalyzer/1.0`.

Para cada dominio procesado, se realizan las peticiones indicadas en la Fig. 2.

- Una vez verificado que el dominio está dado de alta en un servidor DNS, se realiza una petición para solicitar el fichero `robots.txt`, como especifica el protocolo del estándar de exclusión de robots [5]. Esta petición se realiza delegando la gestión de redirecciones HTTP en el cliente utilizado.
- Si el fichero `robots.txt` existe y no posee una entrada que habilite el acceso a la página de inicio del sitio web (`/`), se finaliza el proceso.
- Si el estándar de exclusión de robots permite el acceso al recurso, entonces se realiza una nueva petición HTTP para obtener el recurso `/` del servidor web. En este caso se deshabilita la gestión automática de redirecciones HTTP, para poder almacenar la información de pasos intermedios y número de redirecciones HTTP que hay que realizar para alcanzar el contenido. Ante cada navegación que devuelva como código de respuesta un código de `HTTP redirect`, se genera una nueva petición al servidor a partir del contenido del campo `Location` de la cabecera HTTP de respuesta del servidor.

Si el proceso termina porque el servidor de DNS o HTTP excede el tiempo máximo permitido, se espera 60 segundos antes de realizar un último reintento.

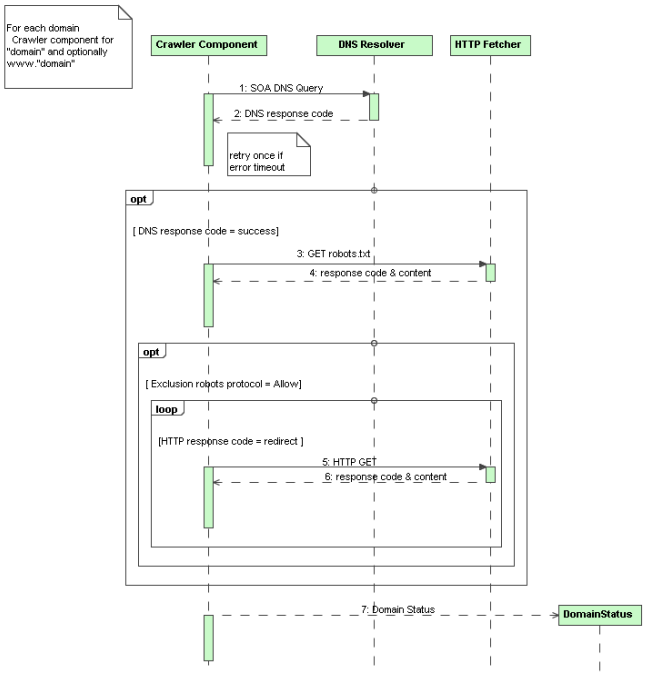


Fig. 2. Secuencia para la Obtención de Información de un Dominio

Es importante tener en cuenta que un nombre de dominio dado de alta en un servidor de nombres no implica necesariamente que deba existir un servidor HTTP para dicho dominio, ni que el servidor HTTP se encuentre escuchando en el puerto por defecto. Por estos motivos, es necesario aclarar que el sistema propuesto sólo considera los servidores HTTP en ejecución en el puerto por defecto (80), y utilizando el protocolo HTTP (no HTTPS ni FTP).

Adicionalmente, es importante destacar que para un nombre de dominio dado, pueden existir múltiples nombres de máquinas en ese dominio con servidor HTTP. El sistema propuesto sólo considerará el servidor HTTP asociado al nombre de dominio y el de la máquina www de ese dominio. Es decir, por cada dominio considerado, el proceso comentado se realizará dos veces, con la excepción de aquellos nombres de dominio que ya comiencen por www, para los que sólo se realizará una única vez (se ha detectado que para muchos dominios es necesario especificar www para acceder a su servidor HTTP; y en otros los servidores en ambos casos proporcionan contenido diferente).

Por otra parte, los clientes HTTP utilizados han sido configurados especificando un tiempo máximo de 240 segundos por petición HTTP, un límite de 15 redirecciones HTTP por petición, un máximo de 5 reintentos ante errores en los que un servidor acepta la petición pero no es capaz de generar la respuesta y deshabilitada la verificación de certificados digitales (aceptar siempre), para maximizar el

número de dominios de los que es posible obtener información.

B. Módulo de Análisis de Crawling

Como resultado del proceso de *crawling* se han obtenido los siguientes datos para cada dominio:

- Tiempo de ejecución total de todo el procesamiento sobre ese dominio.
- Código de respuesta del servidor DNS.
- Nombre del servidor DNS autoritativo.
- Nombre del servidor de correo para el dominio.
- Código de respuesta para la petición HTTP que solicita el fichero `robots.txt`.
- Valor lógico que especifica si se permite el acceso a la página de inicio del sitio, según el estándar de exclusión de robots.
- Contenido del fichero `robots.txt`.
- Código de respuesta para la petición HTTP que solicita la página de inicio del sitio.
- Cabeceras HTTP de la respuesta que contiene la página de inicio del sitio web.
- Contenido de la página de inicio del sitio web.
- Tipo MIME de la página de inicio del sitio web, según lo especificado en la cabecera HTTP de respuesta por el servidor web.
- Información del servidor web, según lo especificado en la cabecera HTTP de respuesta.
- Número de redirecciones HTTP realizadas para alcanzar la página de inicio del sitio.
- Lista de las diferentes peticiones realizadas en el caso de redirecciones, junto con los códigos de respuesta obtenidos en cada caso.
- URL final del recurso devuelto como página de inicio del sitio.

Toda esta información por dominio es utilizada por el módulo de análisis de *crawling* para generar los resultados presentados en la sección III. Para la obtención de los dominios que se encuentran en *parking* es necesario introducir previamente la lista de servidores de nombres del servicio de *parking*.

El listado de dominios .es y de dominios de *parking* utilizados han sido proporcionados por Red.es, mediante un convenio de colaboración con la Universidade da Coruña (ver sección de Agradecimientos).

C. Herramienta de Consulta Web

Se ha implementado una herramienta web que requiere autenticación y permite consultar tanto los datos de entrada del proceso de *crawling* como los de salida. En particular permite consultar:

- La lista de dominios .es analizados.
- La lista de nombres de dominios del servicio de *parking*.
- Los datos recopilados para un dominio en concreto, listando un resumen de los datos para todos los dominios o realizando la búsqueda por nombre para un dominio determinado. Se proporciona también una pantalla de detalle para visualizar toda la información recopilada para un dominio.
- Los resultados estadísticos obtenidos para el *crawling* realizado entre unas fechas.

Adicionalmente, proporciona la funcionalidad para obtener los datos para un dominio concreto en tiempo real.

En la sección III se muestra algún ejemplo de uso de la herramienta.

III. Evaluación Experimental

A. Recursos utilizados

La recolección de páginas y posterior análisis de las mismas se ha realizado en un ordenador Intel(R) Pentium(R) 4 con 2 CPUs a 2.60GHz y 1GB de memoria RAM. El sistema operativo utilizado fue Linux Ubuntu 8.10.

B. Experimentos y Resultados

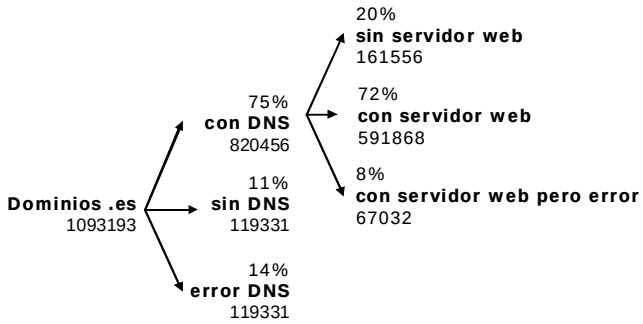
Se ha utilizado el sistema propuesto para realizar el estudio automático de los dominios .es, partiendo del listado facilitado por la Entidad Pública Empresarial Red.es [6] en Diciembre de 2008.

El crawling se ha realizado entre el 27 de Marzo y el 2 de Abril de 2009, con una paralelización de 80 peticiones simultáneas, y durante 5 días.

En la Fig. 3 se muestran los resultados generales obtenidos para los dominios .es analizados.

Han sido procesados 1093193 dominios, de los cuales el 75% estaban dados de alta en algún servidor de DNS. El 14% de los dominios considerados no estaban dados de alta en ningún servidor DNS. En el 11% restante de los dominios la consulta DNS ha finalizado con algún tipo de error desconocido o con *timeout*.

Por otra parte, se ha realizado una asociación entre el nombre del servidor DNS autoritativo para cada dominio y el listado de dominios de servicios de parking, concluyendo que el 8,23% de los dominios .es se encuentran en *parking*.



Dominios con servicio de Parking	8.23%	90048

Fig. 3. Resultado Generales para Dominios .es

Los resultados porcentuales del análisis de los servidores web en los dominios .es se expresan respecto a aquellos dominios que están dados de alta en algún servidor DNS. Tras el análisis se puede concluir que el 72% de los dominios .es que están dados de alta presentan servidor web. Existe un 20% de dominios que no tienen un servidor HTTP instalado y en el restante 8% de los casos tienen servidor web pero han devuelto algún tipo de error al intentar acceder a ellos.

Se ha concluido que en el 80% de los dominios se ha obtenido una respuesta de un servidor web. Además, en el 72% de los casos la respuesta ha sido exitosa (código HTTP

200). Por otra parte, en un 20% de los dominios no es posible acceder al servidor web debido a diferentes tipos de problemas.

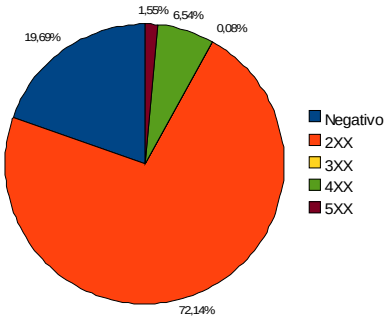


Fig. 4. Resumen de Códigos de Respuesta HTTP

La Fig. 4 muestra un resumen de los diferentes códigos HTTP de respuesta obtenidos. En la Fig. 5 se incluye la descripción detallada de los diferentes códigos de respuesta. Los códigos de error negativos se corresponden con aquellos casos en los que no ha sido posible contactar con el servidor HTTP (negativo mayor que -6). Los códigos de error menores que -6 representan errores graves de redirección en la configuración del sitio web. En ambos casos no es posible acceder a la página destino.

Es importante tener en cuenta que los resultados asumen un solo servidor por dominio. En el apartado II.A.2 se comentó que para cada nombre de dominio se realizaba un máximo de dos peticiones (el nombre de dominio tal cual y la máquina www). Para los cálculos se están considerando ambos resultados como pertenecientes al mismo dominio, utilizando como datos para el dominio la respuesta más exitosa (si el acceso sin www devolvió un error y el acceso con www un código HTTP 200, se considera que el nombre de dominio tiene servidor HTTP).

Descripción	Códigos de Respuesta	# Dominios	Porcentaje
Unknown Error	-1	103250	63,91%
Timeout	-2	8554	5,29%
Unknown Host	-3	44803	27,73%
No Route to Host	-4	1922	1,19%
No HTTP Response	-5	2636	1,63%
Runtime Error	-6	127	0,08%
Redirect Without Location	-10	23	0,01%
Redirect Infinite Loop	-11	241	0,15%
	Negativo	**161556**	**19,69%**
OK	200	591868	100,00%
Created	201	1	0,00%
No Contenet	204	1	0,00%
	2XX	**591870**	**72,14%**
Multiple Choice	300	4	0,57%
See Other	303	645	92,54%
Temporary Redirect	307	48	6,89%
	3XX	**697**	**0,08%**
Bad Request	400	2485	4,63%
Unauthorized	401	1887	3,52%
Forbidden	403	35311	65,85%
Not Found	404	13840	25,81%
Method Not Allowed	405	1	0,00%
Not Acceptable	406	22	0,04%
Request Timeout	408	22	0,04%
Gone	410	51	0,10%
Request-URI Too Long	414	7	0,01%
	4XX	**53626**	**6,54%**
Internal Server Error	500	1055	8,30%
Bad Gateway	502	49	0,39%
Service Unavailable	503	11592	91,23%
Gateway Timeout	504	10	0,08%
Bandwidth Limit Exceeded (Apache)	509	1	0,01%
	5XX	**12707**	**1,55%**
			820456

Fig. 5. Detalle de Códigos de Respuesta HTTP

La Fig. 6 resume el análisis realizado sobre la utilización del estándar de exclusión de robots en los sitios web. El 35% de los dominios con servidor web definen el fichero `robots.txt` y en un 2,38% de los casos no está permitido acceder a la página de inicio del sitio web.

Servidores con robots.txt	35%	206257
Servidores con robots.txt y exclusión	2,38%	14088

Fig. 6. Resultados robots.txt

Otro de los aspectos analizados son las redirecciones HTTP. Cuando un servidor HTTP desea indicar a un cliente que se conecte a otro recurso, el servidor web devuelve como respuesta un código especial (3XX) en función del caso. Para el código 301 (*Moved Permanently*) el servidor envía una cabecera especial en la respuesta (`Location`) con la ruta al nuevo recurso. En la Fig. 5 se muestra el detalle de dominios que han devuelto códigos 3XX. Sin embargo, en ese listado no se están considerando aquellos casos en los que el cliente HTTP fue capaz de redirigir a la página correcta.

En la Fig. 7 se muestran los resultados del análisis de redirecciones. En el 25,17% de los servidores web se ha realizado al menos una redirección HTTP para alcanzar la página de inicio. En el 20,51% de los servidores, las redirecciones permiten alcanzar finalmente la página de inicio.

La Fig. 7 también permite diferenciar entre servidores que redirigen a otra página en su mismo dominio (2,71% de los servidores), los que redirigen al mismo dominio prefijado con www (4,03%), aquellos que redirigen a un protocolo seguro (HTTPS) (0,21%) y los que redirigen a un dominio .com (8,70% de los servidores web de dominios .es redirigen a dominios .com).

Servidores que utilizan HTTP *redirects*	25,17%	149015
Exitosamente	20,51%	121393
Que redirigen al mismo dominio	2,71%	16059
Que redirigen al mismo dominio con www	4,03%	23878
Que redirigen a un protocolo seguro	0,21%	1215
Que redirigen a un dominio .com	8,70%	51486

Fig. 7. Resultados HTTP *redirects*

El sistema de *crawling* debe de contemplar cualquier funcionamiento anómalo de un servidor web, para evitar que el proceso se detenga debido a errores inesperados. Como curiosidad, a continuación se comentan algunos casos especiales detectados:

- Algunos servidores web redirigen de forma indefinida (e.g. www.versapak.es,www.elyfit.es, solovideochat.es,svensk-vodka.es,www.creavida.es)
- Algunos especifican direcciones absolutas para redirecciones, pero sin especificar el protocolo, por lo que se crean URLs incorrectos (e.g. antidot.es, www.trolleybus.es,astrosoleil.es,www.delire.es, www.ferroviaire.es,hifison.es, alibido.es, raffiniti.es, wallinvest.es,www.iamdc.es,www.iamdc.es, ardenninvest.es, fedege.es, domial.es, entre otros).
- Algunos servidores web no terminan de generar la salida, provocando un error de memoria insuficiente (e.g. viajeszaping.es, www.asvi.es).
- Existen algunos casos en los que un navegador funciona correctamente pero el cliente HTTP utilizado no es capaz de determinar el fin de una secuencia de redirecciones (e.g. www.osteolinks.es).

Por último, se ha analizado la información relativa al servidor web que éste devuelve en las cabeceras HTTP, concluyendo que más del 50% de servidores de dominios .es son Apache, un 19% Microsoft Internet Information Server y un 24% no especifican información de servidor.

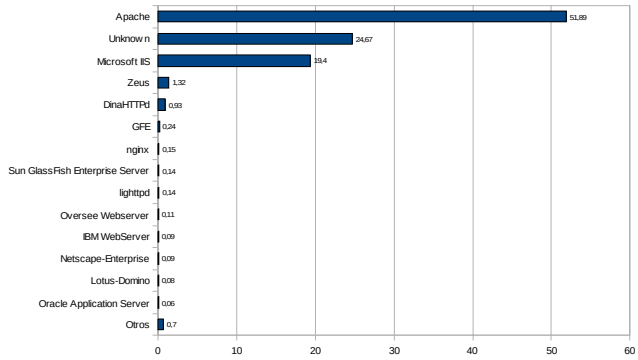


Fig. 8. Uso de Servidores Web

C. *Ejemplos de uso de la Herramienta Web*

En la parte superior de la Fig. 9 se muestra la pantalla principal de la herramienta web para un usuario ya autenticado. En el menú superior se muestran las diferentes opciones existentes que se corresponden con los casos de uso especificados en el apartado II.C.

En la parte central se muestra la pantalla de visualización del resumen de información obtenida para cada dominio, paginada. En particular muestra un enlace a la pantalla de detalle, en la que se visualiza toda la información del dominio, incluyendo su contenido y el de su fichero `robots.txt`, si lo tiene definido. A través del enlace "try" es posible obtener la información del dominio de nuevo, en tiempo real.

IV. CONCLUSIONES

Tras el análisis realizado, se puede concluir que la gran mayoría de los dominios .es se encuentran registrados en algún servidor de nombres (75%) y que a su vez presentan un servidor web (un 80%). El 8,23% de los dominios se encuentran en *parking* y sólo el 2,38% poseen el fichero de exclusión de robots para expresar su deseo de no ser tratados por los sistemas de *crawling*. También se ha comprobado que existe un gran porcentaje de servidores que utilizan HTTP *redirects* para desviar la petición de su página de inicio hacia otras páginas, sitios o protocolos de acceso. Se puede destacar que un 8,70% de los sitios web de dominios .es redirigen a dominios .com.

También es importante resaltar que existen multitud de situaciones excepcionales que un *crawler* tiene que ser capaz de tratar para no resultar "atrapado" por sitios web que por errores de definición intencionados o no, hacen que el acceso a un recurso pueda finalizar con un bucle de redirecciones infinito o una página que no tiene fin.

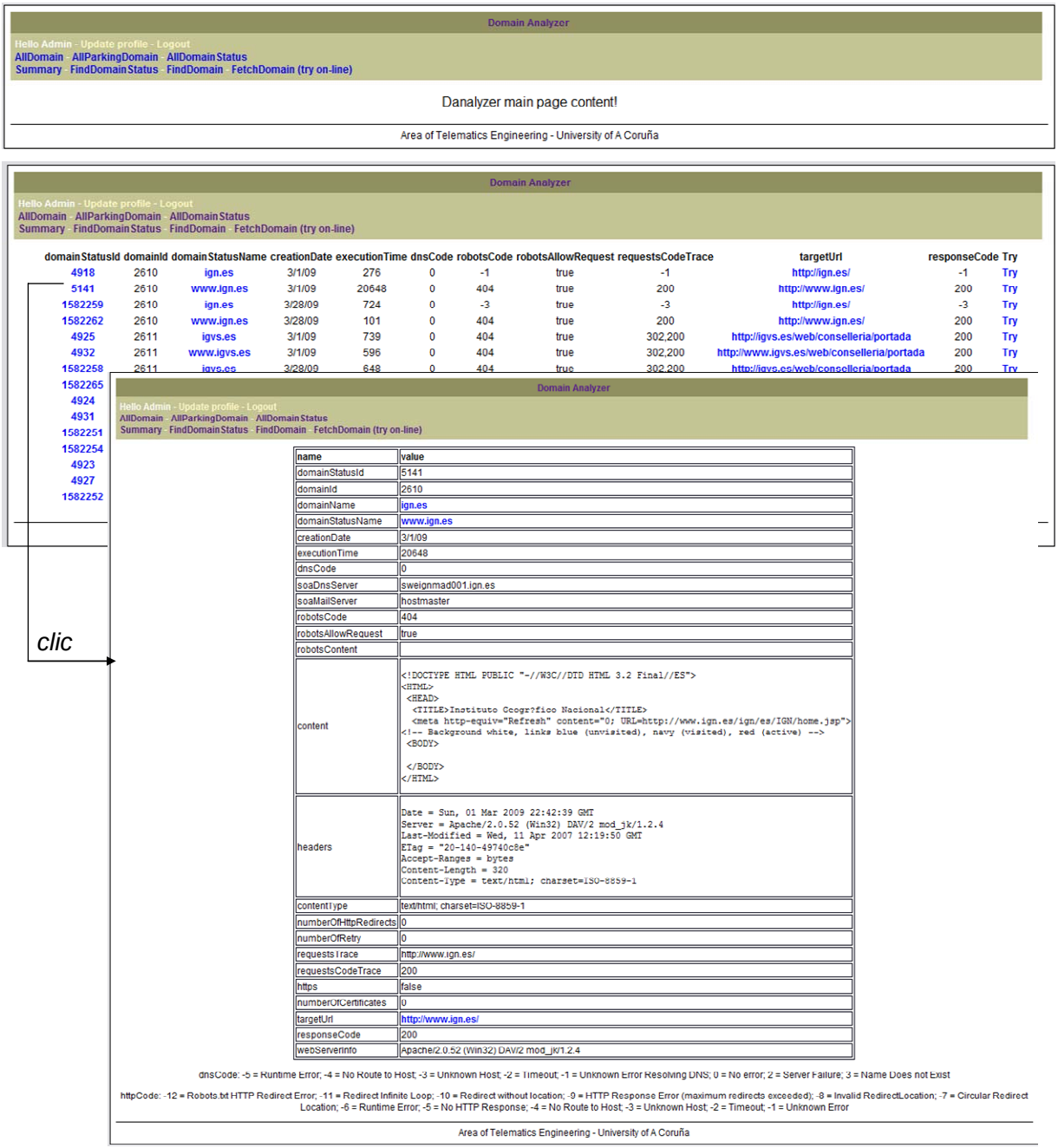


Fig. 9. Herramienta Web

V. Trabajos Futuros

En la segunda fase de este trabajo se realizarán estadísticas de redirecciones no HTTP. Se incluirán entre otros los siguientes aspectos:

- Análisis de redirecciones no HTTP. Además de los HTTP *redirects*, existen otros muchos mecanismos con los que el administrador de un sitio web puede redirigir una petición a otro recurso. Sin embargo, estos mecanismos no son considerados por los sistemas de *crawling* convencionales.
- Análisis de los tipos de documentos referenciados desde las páginas principales de los sitios web analizados
- Análisis de la presencia de formularios en las páginas de inicio.

Agradecimientos

A la Entidad Pública Empresarial Red.es, por proporcionar el listado de dominios .es y dominios del servicio de *parking* a partir del que se ha elaborado este informe.

Referencias

[1] M. Bergman. The Deep Web. Surfacing Hidden Value. *Technical report, BrightPlanet LLC*. Diciembre 2000
[2] Jesse James Garrett. Ajax: A New Approach to Web Applications. *Adaptive Path*. Febrero 2005
[3] DnsJava – http://www.dnsjava.org/
[4] Jakarta Commons HTTPClient – http://hc.apache.org/httpclient-3.x/index.html
[5] The Web Robots Pages – http://www.robotstxt.org/
[6] Entidad Pública Empresarial Red.es – http://www.red.es

Search Shortcuts: recomendación de consultas en buscadores web

Fidel Cacheda, Víctor Carneiro, Diego Fernández, Vreixo Formoso
Departamento de Tecnologías de la Información y las Comunicaciones,
Universidade da Coruña
Campus de Elviña s/n. 15017 A Coruña
fidel@udc.es, viccar@udc.es, dfernandezi@udc.es, vformoso@udc.es

***Resumen*—La recomendación de consultas, o *Query Suggestion*, es una práctica común en los principales buscadores Web, encaminada a mejorar la experiencia del usuario. En este artículo proponemos un nuevo modelo, *Search Shortcuts*, consistente en recomendar consultas que fueron útiles a usuarios que habían buscado información semejante anteriormente. Dicho modelo formaliza el problema de *Query Suggestion*, y permite evaluar los algoritmos empleados a partir de los *query logs* de los buscadores. Hemos evaluado la aplicación de técnicas de filtrado colaborativo, especialmente populares por su éxito en comercio electrónico, en datos procedentes de buscadores reales. Además, hemos estudiado diversas alternativas y soluciones a problemas presentes en este nuevo dominio, como la extracción de información de los *query logs*, o la necesidad de nuevas métricas para evaluar la calidad de las recomendaciones. Los experimentos realizados confirman los buenos resultados de esta aproximación, y abren interesantes perspectivas para futuros trabajos.**

***Palabras Clave*—Search shortcut, Filtrado Colaborativo, Sistemas Recomendadores, Query Suggestion, Evaluación**

I. Introducción

El principal objetivo de un motor de búsqueda es ayudar al usuario a satisfacer sus necesidades de información de un modo eficiente. En este sentido, cualquier ayuda proporcionada al usuario para reducir el tiempo invertido en la búsqueda tiene una gran importancia. De hecho, los principales motores de búsqueda, además de poder responder a las peticiones en unos pocos cientos de milisegundos, normalmente ofrecen al usuario algunas sugerencias, en forma de consultas, que están de algún modo relacionadas con la información que el usuario necesita. Estas *sugerencias* tienen como finalidad dirigir al usuario en la dirección correcta y en la mayoría de los casos ahorrarle tanto tener que realizar un excesivo número de consultas como consumir demasiado tiempo en la búsqueda.

Tanto el diseño como la evaluación de algoritmos efectivos y eficientes para tales sugerencias suponen una tarea compleja. Por ejemplo, en *query suggestion*, tradicionalmente han sido utilizados estudios con usuarios reales para probar el rendimiento de los métodos propuestos en la literatura. Aunque se ha considerado muy precisa la evaluación con usuarios reales, su principal inconveniente es que los experimentos darán resultados diferentes cada vez, lo cual hace difícil una comparación amplia de tales técnicas.

En este trabajo definimos formalmente el *Search Shortcut Discovery Problem* como un problema relacionado con la recomendación de consultas en motores de búsqueda y con las posibles reducciones en la longitud de las sesiones de los usuarios. La idea es recomendar consultas que ya hayan permitido a algún usuario (en el pasado) encontrar con éxito la información deseada usando un proceso de búsqueda similar. Por ejemplo, supongamos que un alto número de usuarios han realizado las consultas q_1, q_2, q_3, y, finalmente, después de enviar la consulta q_4, encontraron la información que necesitaban. Por tanto, podremos considerar que la consulta q_4 es relevante para los usuarios interesados en temas relacionados con q_1, q_2 y q_3. Así, en cuanto otro usuario empiece a buscar temas relacionados con q_1, q_2 o q_3, la consulta q_4 será propuesta como un *shortcut*.

Obviamente, un *shortcut* en la sesión del usuario será más efectivo cuanto antes se sugiera. Además, un *shortcut* para una sesión no tiene por qué anticipar la última consulta, es decir, una consulta propuesta será aceptable siempre que reduzca el tamaño de dicha sesión. Cuanto mayor sea esta reducción, más importante será el *shortcut*. Partiendo de esta idea, definimos una metodología de evaluación para el problema definido basada en *query logs*, que permitirá una comparación sencilla y directa de las técnicas que pueden ser aplicadas al problema.

En particular, hemos estudiado la aplicación de algoritmos de Filtrado Colaborativo. El Filtrado Colaborativo es una técnica de recomendación basada en las preferencias de los usuarios, que ha proporcionado muy buenos resultados en contextos como el comercio electrónico. Su aplicación al *Search Shortcuts Discovery Problem*, y, en general, a cualquier sistema basado en datos provenientes de *query logs*, trae consigo una serie de retos que serán comentados a lo largo de este artículo.

Resumiendo, las principales novedades presentadas en este trabajo son:

- un modelo específico de recomendación: el *Search Shortcuts Discovery Problem*.
- una nueva métrica para evaluar la eficacia de los algoritmos de recomendación aplicados al problema propuesto.
- la aplicación y evaluación de varios algoritmos de Filtrado Colaborativo.

El resto del artículo está organizado de la siguiente manera. En la siguiente sección presentamos un estado del arte. Después, en la sección 3, definimos un modelo teórico para el *Search Shortcuts Discovery Problem* e introducimos una métrica de evaluación. A continuación, evaluamos la aplicación de métodos de Filtrado Colaborativo en la resolución del problema (sección 4). En la sección 5, introducimos los experimentos llevados a cabo y comentamos los resultados obtenidos. Finalmente, presentamos algunas conclusiones y trabajos futuros.

II. Estado del arte

La tarea de proporcionar recomendaciones a usuarios ha atraído el interés de investigadores en los últimos años. En la literatura podemos encontrar dos aproximaciones a este problema: sistemas recomendadores y sistemas de *query suggestion*.

Las técnicas de *query suggestion* tratan el problema de recomendar consultas a usuarios de un motor de búsqueda, y normalmente se basan en datos que proceden de *query logs*. Las técnicas empleadas son muy diversas, abarcando desde las sugerencias triviales de consultas muy repetidas en el pasado, hasta el uso de algoritmos de *clustering* para determinar consultas parecidas [1], o el uso de la información obtenida a partir de los *clicks* en las webs que el usuario visita para medir la similitud entre consultas [2].

Por otro lado, los Sistemas Recomendadores han sido usados en diferentes dominios, siendo especialmente exitosos en comercio electrónico. Pueden ser divididos en dos grandes clases: los basados en filtrado de contenido, y los basados en filtrado colaborativo. También han sido propuestos métodos híbridos, que combinan ambos tipos de sistemas.

Como el propio nombre sugiere, las aproximaciones basadas en filtrado de contenido basan sus recomendaciones en el contenido de los elementos que serán sugeridos. Son ampliamente empleadas con documentos de texto, pero presentan serias limitaciones cuando tratan con contenido multimedia ya que la visión de bajo nivel que una máquina puede llegar a tener del contenido difiere de la visión de alto nivel que tienen los usuarios. Además, y lo que es más importante, sus sugerencias no están influenciadas por la *calidad* de los contenidos desde el punto de vista del usuario, es decir, la opinión (subjetiva) que los usuarios tienen de los elementos. Por otro lado, los algoritmos de *Filtrado Colaborativo* se basan en la experiencia real de usuarios, en las preferencias que éstos dieron en el pasado sobre uno u otro elemento.

Principalmente, existen dos tipos de algoritmos: basados en memoria y basados en modelo. Los basados en memoria utilizan directamente toda la información disponible para identificar los usuarios más similares al actual [3], los elementos más parecidos a aquel sobre el que se quiere hacer la predicción [4], o una combinación de ambos métodos [5]. Generalmente, los algoritmos basados en memoria son bastante simples y producen buenas recomendaciones. Desafortunadamente, como para cada sugerencia se usan todos los datos, presentan serios problemas de escalabilidad. Por otro lado, los algoritmos basados en modelo construyen en primer lugar un modelo para representar el comportamiento de los usuarios previos, permitiendo así predecir de un modo más eficiente sus valoraciones online. Son normalmente más complejos, y la fase de construcción del modelo puede consumir una gran cantidad de tiempo. Además, los modelos son difíciles de configurar, sensibles a los cambios en los datos y altamente dependientes del dominio de la aplicación. En la literatura, las aproximaciones basadas en modelo se sustentan en álgebra lineal [4], en clustering [6], etc.

III. *Search shortcuts*: modelo teórico y métrica de evaluación

A continuación se expondrá el modelo teórico definido para poder aplicar las técnicas de Filtrado Colaborativo en la recomendación de consultas, así como la métrica propuesta para evaluar cómo se ajustan las recomendaciones a las consultas reales (extraídas de un *query log*). Para conseguir dichas recomendaciones se hace uso de algoritmos indicados en la sección anterior.

Sea $\mathcal{U} = \{u | u \text{ es un usuario}\}$ el conjunto de todos los usuarios de un motor de búsqueda. Sea $\mathcal{Q} = \{q | q \text{ es una consulta de un usuario}\}$ el conjunto de todas las consultas que han sido enviadas por los usuarios de un motor de búsqueda.

Definición 1: Una sesión para un usuario $u \in \mathcal{U}$ es una secuencia de consultas que el usuario u ha enviado a un servicio de búsqueda con el objetivo de satisfacer una necesidad de información. Formalmente, $\sigma^u =< q_1^u \dots q_n^u >$. Se asume que todas las consultas están relacionadas con la misma tarea de búsqueda.

Se eliminará el superíndice u en la especificación de σ, e.g. $\sigma =< q_1 \dots q_n >$, cuando esté claro u en el contexto a tratar.

El i-ésimo elemento de una sesión se denota como σ_i.

$\mathcal{S}$ representa el conjunto de todas las sesiones.

Definición 2: Definimos una función $c : \mathcal{S} \times [1..n] \rightarrow \{0, 1\}$ como $c(\sigma, i) = 1$ si en σ el usuario ha hecho click en al menos una de las URLs mostradas como resultado para σ_i.

Definición 3: Una sesión σ será ***satisfactoria*** si, y sólo si, $c(\sigma, n) = 1$; ***insatisfactoria*** en otro caso.

Definición 4: Definimos un ***shortcut*** **de grado** k como una función $h : \mathcal{S} \rightarrow 2^{\mathcal{Q}}$ que toma como argumento una sesión y devuelve un conjunto de consultas de cardinalidad menor que k, i.e. $|h(\sigma)| \leq k$.

$\mathcal{H}$ es el conjunto de todas las posibles funciones *shortcut*.

Definición 5: La **cabeza** $\sigma_{t|}$ de σ **hasta** $t \leq n$ es la secuencia de las primeras t consultas en σ, i.e. $\sigma_{t|} =< q_1, \dots, q_t >$

Definición 6: La **cola** $\sigma_{|t}$ de σ **desde** $t \leq n$ es la secuencia de las últimas $n-t$ consultas en σ, i.e. $\sigma_{|t} =< q_{t+1}, \dots, q_n >$

Definición 7: Sea σ una sesión satisfactoria. La similitud de un *shortcut* de grado k, h, sobre una cabeza $\sigma_{t|}$ y una cola $\sigma_{|t}$ se define como

$$s\left(h\left(\sigma_{t|}\right), \sigma_{|t}\right) = \frac{\sum_{q \in h(\sigma_{t|})} \sum_{m=1}^{n-t} \left[q = \left(\sigma_{|t}\right)_m\right] f(m)}{|h(\sigma_{t|})|} \quad (1)$$

Donde $f(m)$ es una función monótona creciente. La función $[q = \sigma_m] = 1$ si, y sólo si, la consulta q es igual a la consulta σ_m.

La función de similitud definida en (1) puede ser usada como una medida de evaluación objetiva para el *search shortcut discovery problem*. Tiene en cuenta tanto el número de consultas recomendadas que forman parte de la sesión real, como su posición en esta, siendo mejor valorado un algoritmo que recomienda las últimas consultas en la sesión. Destacar que la función de similitud puede ser reescrita incluyendo la función c, lo cual permite dar importancia únicamente a aquellas consultas que tuvieron algún resultado sobre el que se ha hecho click.

Es importante destacar que la principal diferencia entre *Query Shortcuts* y *Query Suggestion* se refleja en la función $\left[q = \left(\sigma_{|t}\right)_m\right]$ en la ecuación (1). Reemplazando el requisito de igualdad estricta $=$ por una relación de similitud, i.e.

$\left[q \sim \left(\sigma_{|t}\right)_m\right]$ (es decir, $\left[q \sim \left(\sigma_{|t}\right)_m\right] = 1$ si, y sólo si, la consulta q es similar a la consulta σ_m) el problema se reduce, básicamente a un problema de *Query Suggestion*. Por tanto, definiendo simplemente una función de similitud adecuada en (1) se podría evaluar también la eficacia de algoritmos de *Query Suggestion*.

Finalmente, deberíamos considerar la importancia de la función $f(m)$. De hecho, según la f escogida, se estarán probando diferentes características de los algoritmos evaluados. Por ejemplo, si consideramos $f(m)$ una función constante $f(m) = c$, estaremos midiendo simplemente el número de consultas en común entre el conjunto de consultas recomendadas y aquellas que el usuario envió. Para dar más importancia a las consultas cuanto más tarde se realicen dentro de la sesión, podemos pensar en emplear funciones no constantes. Así, por ejemplo, una función exponencial $f(m) = e^m$ les daría mucha más relevancia a aquellas consultas que estuviesen más cercanas al final de la sesión. Se pueden usar funciones f más suaves para moderar el efecto de la posición de las consultas dentro de la sesión.

IV. Filtrado Colaborativo aplicado a Search Shortcuts

El problema de *Query Shorcuts* es, sin lugar a duda, un problema de recomendación, por lo que el uso de técnicas de Filtrado Colaborativo se justifica debido a su relevancia en la resolución de otros problemas de recomendación similares. En tales técnicas están presentes un conjunto de usuarios, $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$, y un conjunto de elementos, $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$. El algoritmo tiene como finalidad predecir la utilidad que un elemento dado tiene para el usuario (tarea de anotación en contexto), o recomendar una lista de elementos (tarea de búsqueda de buenos elementos). Al usuario para el que se está realizando la predicción (o recomendación) en un instante determinado se le denomina usuario activo, u_a.

Para llevar a cabo su tarea, el Filtrado Colaborativo se basa en las preferencias de los usuarios. Dichas preferencias son tenidas en cuenta como valoraciones de un elemento, que no son más que valores numéricos que representan la utilidad de un elemento para un usuario dado. El subconjunto de todas las valoraciones se conoce como R. Las valoraciones pueden ser introducidas explícitamente por los usuarios, o bien extraídas implícitamente de la interacción de los usuarios con el motor de búsqueda (p.e. de los datos del *query log*). Las preferencias de todos los usuarios se almacenan en una matriz de usuarios-elementos, conocida con el nombre de matriz de valoraciones, V. Un valor de V, $v_{ui} \in R \cup \{\varnothing\}$, hace referencia a la valoración que el usuario $u \in \mathcal{U}$ ha dado al elemento $i \in \mathcal{I}$. El valor $\{\varnothing\}$ indica que el usuario todavía no ha valorado el elemento.

Para aplicar Filtrado Colaborativo al problema de *Search Shortcuts*, los valores de la matriz han de ser extraídos de la información disponible en los datos del *query log*. Sin embargo, esta no es una tarea sencilla, y en ella surgen una serie de problemas entre los que destacamos dos.

En primer lugar, los conceptos del problema de *Search Shortcuts* (usuarios, consultas, términos y sesiones) tienen que ser mapeados al problema de Filtrado Colaborativo puro (usuarios y elementos). Como el objetivo en el problema de *shortcuts* es recomendar consultas, parece razonable tratar cada sesión como un *usuario* y cada consulta como un *elemento*.

En segundo lugar, las valoraciones han de ser inferidas a partir de la información del *query log*. Como una primera aproximación, en este trabajo valoramos las consultas centrándonos en la última de cada sesión. Si dicha última consulta fue exitosa (el usuario hizo un click sobre al menos uno de los resultados), entonces la consulta recibe una valoración positiva en esa sesión. En caso contrario, se le da una valoración negativa. El resto de consultas se consideran neutrales.

Finalmente, hay también ciertas diferencias significativas entre las características de los datos de un *query log* y los de los *datasets* usados por los recomendadores basados en Filtrado Colaborativo tradicional. Estos *datasets* son mucho más densos que un *query log*, es decir, tienen muchas más relaciones (valoraciones) entre usuarios y elementos. Por ejemplo, en un sitio web de comercio electrónico la mayoría de los productos han sido valorados o comprados por diversos clientes. De este modo, podemos obtener información sobre un elemento a través de varios usuarios. Sin embargo, en los *query logs* existen muchas consultas que aparecerán en una única sesión, y, así, no podremos confiar en su utilidad. Esta falta de información se corresponde con el conocido problema de la baja densidad de información [7] que afecta a la mayoría de los algoritmos de Filtrado Colaborativo. Un buen rendimiento en entornos de baja densidad de información es un requisito que un algoritmo debe cumplir para ser aplicado exitosamente al problema de *Search Shortcuts*.

Además, diferentes alternativas pueden ser aplicadas para identificar consultas únicas en el *query log*. Una primera aproximación consiste en tratar la cadena de búsqueda (la consulta) como un todo, considerando de este modo que dos consultas son la misma sólo si son idénticas. El problema de esta aproximación es que consultas con exactamente los mismos términos, pero en diferente orden, no serán consideradas iguales, disminuyendo así la densidad de información en el *dataset*. Por tanto, una mejor aproximación debería tener en cuenta los términos que componen una consulta, reordenándolos adecuadamente si fuese necesario. En la práctica, técnicas usadas habitualmente en Recuperación de Información, como *Term Stemming* o *Stopwords Removal*, pueden ser usadas también para aumentar la densidad de información en la matriz.

Finalmente, los *query logs* también contienen muchos más datos que los que se pueden encontrar en los dominios tradicionales. Mientras los usuarios de un sitio web de comercio electrónico pueden ser más o menos estáticos, nuevas sesiones y consultas son registradas continuamente en un *query log*. Para aplicar Filtrado Colaborativo al problema de *Search Shortcuts*, necesitamos algoritmos computacionalmente eficientes tanto en tiempo de predicción como de entrenamiento, ya que el modelo debe ser continuamente actualizado.

V. Experimentos

A. Configuración de los Experimentos

Los distintos algoritmos han sido evaluados usando datos de *query logs* reales, de los motores de búsqueda AOL

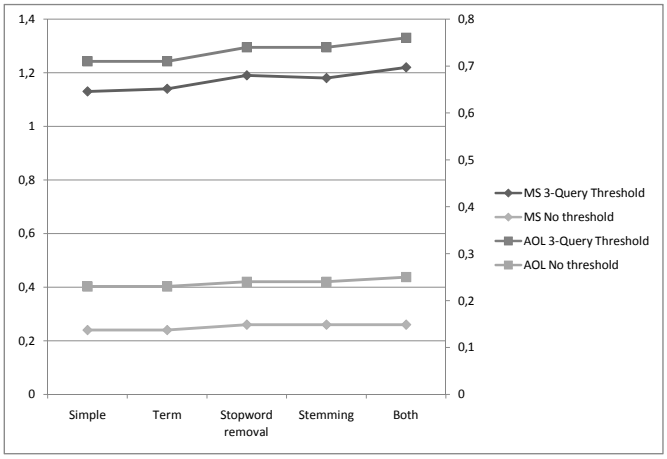


Fig. 1. Efecto de las diferentes técnicas de preprocesado sobre los *query logs* de AOL -eje izquierdo- y MSN -eje derecho-. Los resultados se muestran en términos de densidad relativa x10^5

[8] y MSN[1]. Hemos derivado diez nuevos *datasets* a partir de los dos ya citados, aplicando diferentes técnicas para el preprocesamiento del *log*. Para comenzar, hemos seleccionado las consultas únicas haciendo uso de las técnicas discutidas en la sección IV:

- Considerar la consulta como una cadena única.
- Tener en cuenta diferentes términos en una consulta.
- Aplicar *stopword removal*.
- Aplicar *term stemming*.
- Aplicar *stopword removal* y *term stemming*.

Además, también hemos estudiado el efecto de considerar sólo sesiones con más de un número determinado de consultas. Ambas técnicas, como hemos visto en la Fig. 1, permiten aumentar la densidad de los datos, lo cual se prevé que mejore la precisión de los resultados. En especial, ignorar sesiones con pocas consultas tiene un impacto significativo en la densidad del *dataset*. Cuando se consideran únicamente sesiones de al menos tres consultas, la densidad aumenta significativamente, lo cual es fácilmente explicable por el hecho de que un gran número de sesiones desaparecen del *dataset* final. En realidad, el número total de sesiones se reduce hasta un 80% en el *dataset* de MSN, y hasta un 70% en el caso del AOL.

Del mismo modo, aplicar cualquier técnica de preprocesamiento también ayuda a aumentar la densidad de datos, aunque esto tenga menos impacto que descartar sesiones con pocas consultas. En este caso la densidad se incrementa debido a que estas técnicas reducen el número de consultas únicas. Los mejores resultados se obtienen aplicando *stopword removal* y *term stemming*, especialmente si las sesiones pequeñas también se descartan.

Para llevar a cabo la evaluación, hemos dividido los datos en dos subconjuntos: entrenamiento y evaluación, siguiendo un proceso en dos pasos. Primero, hemos escogido de modo aleatorio un porcentaje de las sesiones para ser usadas como datos para entrenar a los algoritmos. Después, alimentamos los algoritmos con las primeras dos consultas de cada una de las sesiones del conjunto de evaluación. Éstas representan la *cabeza de la sesión* que el algoritmo necesita como contexto de predicción.

Los resultados son entonces evaluados usando la medida de similitud propuesta en la sección III, especialmente diseñada para el problema de *Search Shortcuts*. Hemos usado diversas variaciones de la función monótona creciente $f(m)$ (ver ecuación 1), para evaluar diferentes aspectos de cada algoritmo:

- Una función constante, $f(m) = 1$
- Una función lineal, $f(m) = m$
- Una función cuadrática, $f(m) = m^2$
- Una función exponencial, $f(m) = e^m$

Para hacer una evaluación más completa, hemos usado también las métricas tradicionales. Estas métricas se clasifican en métricas de precisión de la predicción y en métricas de precisión de la clasificación.

Las métricas de precisión de la predicción miden la diferencia entre la valoración que el sistema predice y la valoración real hecha por el usuario. Entre ellas, hemos usado el Mean Absolute Error (MAE), ya que es la métrica más popular en la literatura, así como nuevas métricas como Good Predicted Items MAE (GPIM) y Good Items MAE (GIM) [9]. Éstas sólo tienen en cuenta los errores de predicción en elementos relevantes, y, por tanto, proporcionan unos resultados más cercanos al punto de vista del usuario.

Por otro lado, las métricas de precisión de la clasificación tienen como objetivo evaluar la calidad de la lista de recomendaciones. Hemos usado tanto *Precision and Recall* como *ROC Curves* [10]. Éstas evalúan un algoritmo basándose en si los elementos recomendados son realmente buenos. También son de utilidad las métricas que tienen en cuenta la ordenación de los elementos en la lista, como *Half-Life Utility* [11].

Finalmente, hemos considerado importante tener en cuenta el *coverage* del algoritmo, es decir, el porcentaje de elementos para los que es capaz de hacer un predicción. Algoritmos con poco *coverage* tenderán a recomendar siempre los mismos elementos y, por tanto, son menos interesantes desde el punto de vista de los *Search Shortcuts*.

B. Algoritmos evaluados

- Item-Mean. Es un algoritmo extremadamente sencillo, que siempre devuelve como predicción la media del elemento en cuestión.
- Algoritmos basados en usuario [3], [12]. Una de las estrategias de filtrado colaborativo más populares. Predicen la utilidad de un elemento mediante un proceso que incluye tres fases:
  1) Calcular la similitud entre el usuario activo y el resto de usuarios. Existen distintas técnicas para tal fin, entre las que hemos evaluado el coeficiente de correlación de Pearson [13], Constrained Pearson [3], Mean Squared Diff [3] y Weighted Pearson [14].
  2) Seleccionar un subconjunto de los usuarios según su similitud con el usuario activo (es decir, sus vecinos). Hemos evaluado la técnica más popular, *maximum number of neighbors* [13], ya que es aquella que ofrece mejores resultados [12].

[1]Este *dataset* hace referencia al *dataset* Microsoft 2006 RFP proporcionado por la conferencia Web Search Click Data 2009

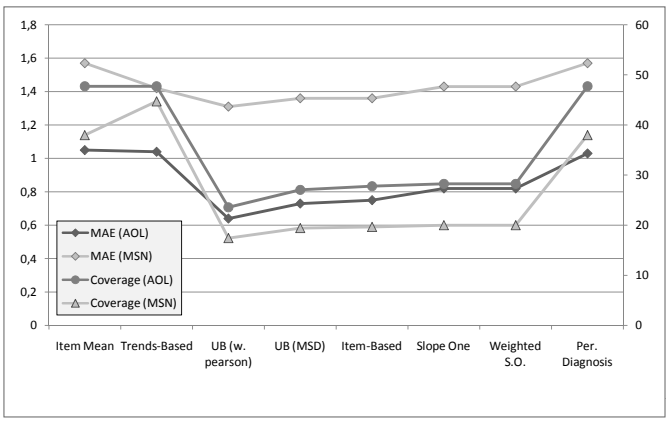


Fig. 2. Resultados en MAE -eje de la izquierda- y coverage -derecha- en los *datasets* AOL y MSN

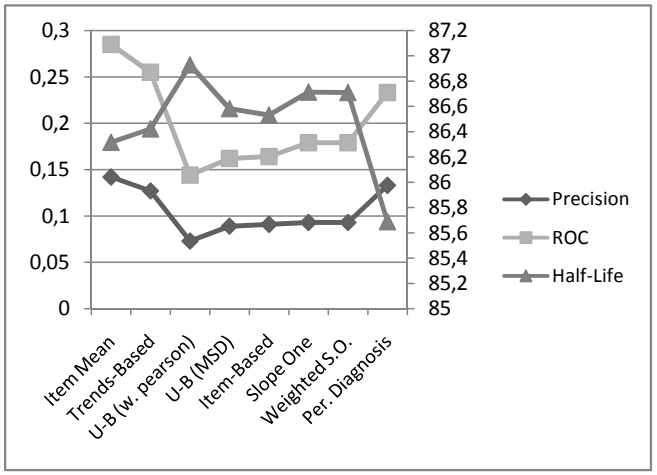


Fig. 3. Resultados, en el *dataset* AOL, en las métricas de precisión, ROC -eje de la izquierda- y half-life utility -derecha-

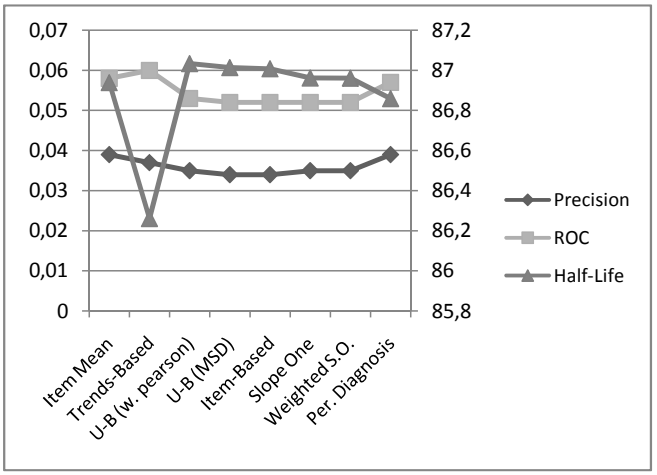


Fig. 4. Resultados, en el *dataset* MSN, en las métricas de precisión, ROC -eje de la izquierda- y half-life utility -derecha-

3) Finalmente, calcular la predicción a partir de las valoraciones de los vecinos. Hemos estudiado dos alternativas: *weighted by correlation* [13] y *Z-Score normalization* [12].

- Algoritmos basados en elementos [7]. Son similares a los basados en usuarios, pero en lugar de buscar los usuarios más parecidos, intentan encontrar elementos similares a aquel para el que queremos realizar la predicción.
- Slope-One [15]. Un sencillo modelo para filtrado colaborativo, basado en las diferencias entre valoraciones. Hemos evaluado las variantes *simple* y *weighted*.
- Personality Diagnosis [16]. Está basado en un modelo probabilístico sencillo que aproxima la forma en que los usuarios valoran elementos mediante una distribución normal.
- Trends-Based [9]. Una aproximación novedosa, basada en las variaciones naturales entre diferentes usuarios y elementos.

C. Resultados

En primer lugar, hemos analizado la precisión en la predicción de los diferentes algoritmos, para así evaluar qué tal se comportan las técnicas de filtrado colaborativo en contextos caracterizados por la baja densidad y gran volumen de datos, como aquel que ahora nos ocupa. Como ya hemos comentado, los dominios en que se han aplicado tradicionalmente estas técnicas son relativamente densos. En la Fig. 2 podemos observar los resultados, según la métrica MAE. Como vemos, la mayoría de algoritmos presentan unos resultados bastante buenos, especialmente en el *dataset* AOL, con un error absoluto medio entre el 5% y el 10%. Sin embargo, estos buenos resultados deben ser interpretados cuidadosamente.

De hecho, si observamos el *coverage* de cada algoritmo (también mostrado en la Fig. 2), podemos ver que los buenos resultados en MAE (es decir, predicciones bastante exactas) están a menudo relacionados con un bajo *coverage*. La razón es que algunos algoritmos extraen muy poca información de los datos disponibles y, por tanto, sólo pueden predecir la utilidad de aquellos elementos para los cuales existe gran cantidad de información en el *dataset*. Tal caso suele corresponder a elementos populares, generalmente "fáciles de predecir", y por tanto la precisión es mejor en ellos. Sin embargo, el usuario normalmente ya los conoce, con lo que la recomendación no es muy útil [17] para él. Este problema suele existir en los algoritmos basados en usuarios o elementos, ya que al tener en cuenta únicamente ciertas partes de la información disponible (las relaciones entre usuarios o elementos), presentan un *coverage* muy bajo en *datasets* con baja densidad de información. Por otra parte, algunas aproximaciones basadas en modelo, como Trends-Based o Personality Diagnosis presentan, globalmente, buenos resultados.

Además, la métrica MAE tiene otra limitación importante en este contexto, relacionada con la forma en que asignamos valoraciones de forma implícita a partir del *dataset*. Teniendo en cuenta que hemos asignado una valoración neutral a todas las consultas en una sesión, excepto la última, y además solamente estamos evaluando en sesiones relativamente grandes, la mayor parte de las consultas van a tener una valoración neutral. Por tanto, para un algoritmo va a resultar más sencillo obtener una predicción para dichos elementos. Por ejemplo, un algoritmo que se limite a predecir siempre una valoración neutral, obtendría resultados muy precisos en la mayoría de el-

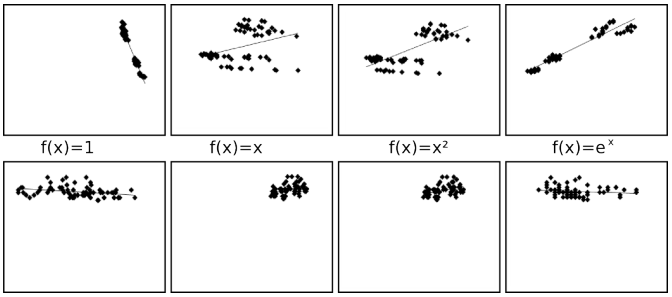


Fig. 5. Correlación entre la Search Shortcuts Similarity y la precisión, dependiendo de la función f elegida, tanto en el *dataset* AOL -arriba- como el MSN -abajo-.

ementos, y estos compensarían los malos resultados obtenidos en otros, ya que la métrica MAE tiene en cuenta la media del error cometido en todo el conjunto de evaluación. De hecho, si estudiamos los resultados obtenidos en las métricas GIM y GPIM, podemos ver que el error aumenta hasta cerca del 50% en el *dataset* MSN, y por encima del 30% en el AOL. Es decir, cuando sólo tenemos en cuenta los elementos buenos (o aquellos predichos como buenos), los resultados empeoran considerablemente.

La misma conclusión puede obtenerse a partir de la evaluación basada en métricas de precisión en la clasificación (los resultados para los *datasets* AOL y MSN pueden verse, respectivamente, en las Fig. 3 y 4). Aunque ningún algoritmo ofrece unos resultados destacados, debemos tener en cuenta las limitaciones de las métricas de evaluación, especialmente cuando esta se realiza en un *dataset* offline, con gran volumen de datos pero baja densidad. En estas condiciones, es muy probable que las recomendaciones ofrecidas por el algoritmo no se correspondan realmente con ninguno de los elementos que tenemos en el conjunto de evaluación. Por tanto, la evaluación no podrá llevarse a cabo de forma satisfactoria. Para minimizar este problema, hemos restringido los elementos que el algoritmo podría recomendar a aquellos presentes en el conjunto de evaluación. Tal hecho debe tenerse en cuenta a la hora de interpretar los resultados obtenidos en estas métricas. Dado que el algoritmo solamente puede recomendar unos pocos elementos, un pequeño error puede llegar a tener un gran impacto en la precisión final.

Además, las métricas tradicionales se basan en el concepto de "relevancia" de un elemento, que es en cierto modo ambiguo en el contexto de *Search Shortcut*. Por supuesto, una consulta debería considerarse relevante cuando conduce al usuario a la información que estaba buscando, pero esto no puede saberse a partir de los datos en el *query log*. Por tanto, los investigadores pueden elegir entre varias formas de interpretar la "relevancia" de una consulta: si tras ella ha finalizado la sesión, el número de resultados que han sido visitados por el usuario, el tiempo que este tarda en consultarlos, etc.

La métrica *Search Shortcut Similarity* que hemos propuesto en la sección III, elimina esta ambigüedad, al definir claramente lo que debemos evaluar: el número de consultas que la recomendación ha ahorrado al usuario. Centrémonos ahora en esta métrica.

En primer lugar, hemos estudiado la relación entre esta nueva métrica y las métricas tradicionales de precisión en la clasificación. En concreto, es especialmente interesante estudiar la correlación entre nuestra métrica y la medida tradicional de precisión, ya que las dos pretenden medir características similares. Como podemos ver en la Fig. 5, la correlación entre ambas métricas depende de la función f elegida. En concreto, si elegimos la función constante, $f(x) = 1$, nuestra métrica está fuertemente correlacionada con la precisión. Esto no es de extrañar, ya que al elegir una función constante mide el porcentaje de consultas en el subconjunto de evaluación que aparecen en la sesión original, es decir, las consultas "relevantes" desde el punto de vista de *query shortcuts*. Esta característica es realmente importante, ya que significa que los resultados de nuestra métrica pueden interpretarse en términos de "precisión". Pero además, al contrario que la métrica de precisión tradicional, nuestra medida de similitud ofrece una interpretación clara de lo que significa relevancia en el contexto tratado, y, además, cómo debe ser calculada. Esto, sin duda, tiene un gran valor a la hora de evaluar y comparar algoritmos entre sí.

Por otra parte, si elegimos una f distinta, podemos evaluar otras características de los algoritmos, que las métricas tradicionales no son capaces de cubrir. Por ejemplo, usando la función exponencial, $f(x) = e^x$, estaremos evaluando no sólo si las consultas recomendadas son "relevantes", sino hasta que punto lo son; es decir, lo cerca que están del final de la sesión, y por tanto el tiempo que potencialmente podrían ahorrar al usuario.

Algoritmo	$f(x) = 1$	$f(x) = x$	$f(x) = x^2$	$f(x) = e^x$
Item Mean	1.956	0.6	0.354	0.267
Trends-Based	1.989	**0.612**	**0.361**	0.265
U-B(w.pearson)	**2.284**	0.607	0.338	0.246
U-B(MSD)	2.181	0.589	0.33	0.252
Item-Based	2.173	0.586	0.329	0.253
Slope-One	2.182	0.589	0.331	0.252
Weighted S.O.	2.181	0.589	0.331	0.252
Per. Diagnosis	1.951	0.598	0.351	**0.272**

Tabla I
RESULTADOS DE VARIOS ALGORITMOS SEGÚN LA MÉTRICA *Search Shortcuts Similarity*, EN EL *dataset* AOL

Algoritmo	$f(x) = 1$	$f(x) = x$	$f(x) = x^2$	$f(x) = e^x$
Item Mean	1.491	0.301	0.165	0.258
Trends-Based	1.416	**0.33**	**0.188**	**0.273**
U-B(w.pearson)	**1.537**	0.268	0.142	0.261
U-B(MSD)	1.505	0.274	0.146	0.265
Item-Based	1.505	0.272	0.145	0.264
Slope-One	1.513	0.27	0.144	0.263
Weighted S.O.	1.513	0.27	0.144	0.262
Per. Diagnosis	1.457	**0.33**	**0.186**	0.263

Tabla II
RESULTADOS DE VARIOS ALGORITMOS SEGÚN LA MÉTRICA *Search Shortcuts Similarity*, EN EL *dataset* MSN

Una evaluación de los distintos algoritmos empleando esta métrica puede consultarse en las Tablas 1 y 2. Los resultados en ambos *datasets* muestran claras diferencias entre los algoritmos basados en memoria y los basados en modelo. Cuando elegimos una f constante, por tanto considerando de igual forma cualquier "shortcut" relevante, independientemente de

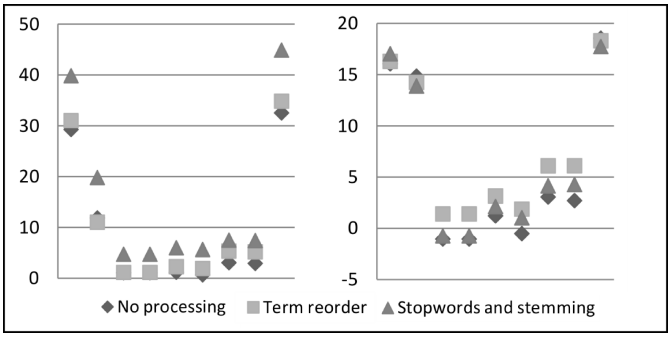


Fig. 6. Porcentaje de mejora, en precisión del algoritmo, cuando las sesiones con menos de 3 consultas son ignoradas, en los *datasets* AOL -derecha- y MSN -izquierda-. El orden de los algoritmos es el mismo que en las Fig. 2, 3 y 4

su posición en la sesión original, los algoritmos basados en usuario obtienen los mejores resultados. Sin embargo, si reemplazamos la f para evaluar hasta qué punto las recomendaciones son útiles para reducir la sesión, podemos comprobar que tanto Trends-Based como Personality Diagnosis presentan resultados más precisos. Además, esto se puede observar en ambos *datasets*, y con la mayoría de métricas. Estos dos algoritmos muestran también buenos resultados en términos de exactitud en la predicción, precisión e incluso *coverage*, por lo que parecen buenos candidatos para el problema de *search shortcuts*. Sin embargo, siguen siendo sensibles a la baja densidad de información, aunque en menor medida que los algoritmos basados en memoria.

De hecho, el negativo impacto que tiene la baja densidad del *dataset* puede observarse en los resultados de los algoritmos con todas las métricas estudiadas. Este comportamiento es el esperado, ya que los algoritmos de filtrado colaborativo estudiados han sido diseñados para contextos con mucha más información. Por tanto, es de esperar que la aplicación de técnicas para reducir la densidad del *dataset*, como las presentadas en la sección IV, tenga un impacto importante en los resultados. Hemos estudiados dos técnicas:

- Ignorar sesiones pequeñas, es decir, sesiones con menos de 3 consultas.
- Aplicación de técnicas de preprocesado: *term reordering*, *term stemming*, y *stopword removal*.

En la Fig. 6 podemos ver que, efectivamente, ignorar sesiones con menos de 3 consultas tiene un gran impacto en la precisión de los algoritmos. En ciertos algoritmos basados en modelo, la mejora obtenida es superior al 30%, y llega hasta cerca del 50% cuando se combina con técnicas de preprocesado. Este resultado es el esperado, ya que los algoritmos de filtrado colaborativo no se comportan bien con usuarios que han valorado pocos elementos, lo que es equivalente a sesiones pequeñas en el contexto de *Search Shortcuts*. Esto corresponde al problema de *Cold-Start* [18], que, como podemos ver, puede solucionarse fácilmente en este contexto. Por otra parte, la mejora no es tan espectacular en los algoritmos basados en memoria (en el centro de ambas gráficas), lo que puede explicarse por el hecho que que estos algoritmos tienen importantes limitaciones para extraer información de *datasets* poco densos. La contribución de este tipo de técnicas a aumentar la densidad del *dataset* (ver Fig. 1) no es suficiente para estos algoritmos. De hecho, en ciertos casos puede incluso llegar a empeorar ligeramente los resultados.

Por otra parte, también hemos estudiado el impacto de las técnicas de preprocesado por sí solas, sin eliminar sesiones pequeñas. Los resultados, como se muestra en la Fig. 7, dependen en gran medida del *dataset* empleado. En el AOL, sólo *term reordering* parece mejorar la precisión de los algoritmos, teniendo las demás técnicas un impacto negativo. Sin embargo, en el *dataset* MSN, tanto *term reordering* por sí solo como combinado con *stopword removal* ofrecen una mejora en la precisión, que llega hasta cerca del 10%, lo que es un gran resultado. De las técnicas estudiadas, sólo *stopword removal*, cuando no se combina con otras técnicas, ofrece siempre malos resultados. En cualquier caso, parece evidente que las técnicas de preprocesado son mucho más útiles cuando se combinan con la eliminación de sesiones pequeñas.

VI. Conclusiones

En este artículo hemos propuesto un nuevo modelo para recomendar consultas a los usuarios de un motor de búsqueda, basado en *Search Shortcuts*. Directamente asociado con este, hemos presentado un modelo bien definido y un *framework* de evaluación, que permite la comparación de algoritmos empleando un *dataset* offline.

En concreto, hemos estudiado la aplicación de técnicas de filtrado colaborativo a este nuevo problema. La evaluación ha sido realizada usando datos reales de *query logs* de los *datasets* AOL y MSN. Hemos obtenido resultados bastante precisos, lo que demuestra que este tipo de algoritmos encaja bastante bien en este problema. Sin embargo, también hemos observado el bajo *coverage* de algunos algoritmos, lo que está relacionado con la baja densidad de este tipo de *datasets*. Sobre todo, las técnicas tradicionales basadas en usuarios o elementos obtienen muy malos resultados en este aspecto, debido a que sólo aprovechan una pequeña parte de la información disponible.

También se han estudiado varias técnicas para reducir el impacto de esta baja densidad. Hemos visto como algunos de estos métodos tienen un gran impacto en los resultados, dando lugar a mejoras significativas. En concreto, ignorar las sesiones con pocas consultas ha demostrado ser una técnica muy exitosa, mejorando la precisión de los algoritmos hasta en un 50%. De la misma forma, técnicas de preprocesado como *term stemming* o *stopword removal* también son útiles, especialmente cuando se combinan con la anterior. Además, las mejoras se producen independientemente del *dataset* empleado.

Por otra parte, cuando se emplean otras métricas en lugar del error en la precisión, los resultados son modestos. Hemos relacionado este hecho tanto con la naturaleza de algunos algoritmos, como con el gran volumen y baja densidad de los datos, así como con las limitaciones de las métricas y metodologías de evaluación tradicionales. Las métricas de precisión en la clasificación obtienen resultados interesantes, pero sólo si se tienen en cuenta los detalles de la metodología de evaluación empleada. En concreto, el uso de *datasets* offline y poco densos impone varias limitaciones, ya que los elementos considerados relevantes por el algoritmo no pueden ser comparados con datos reales en muchos casos.

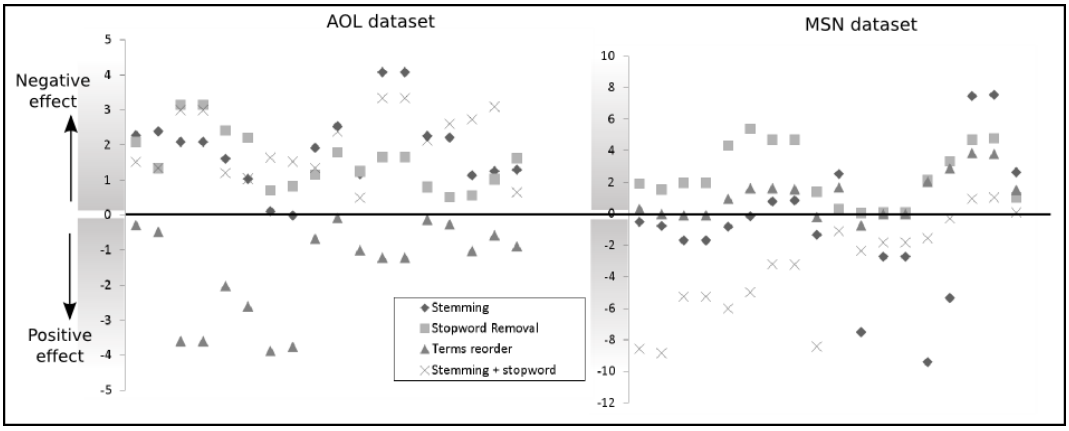


Fig. 7. Efecto de distintas técnicas de preprocesado en la precisión, en los *datasets* AOL y MSN. Los resultados están expresados en términos de porcentaje de variación respecto a la precisión sin aplicar ninguna de dichas técnicas.

Para superar estos problemas hemos propuesto una nueva métrica, especialmente diseñada para la evaluación del problema de *Query Shortcuts*. Su principal ventaja es que los resultados pueden ser fácilmente interpretados en términos de cuán buenas son las recomendaciones del algoritmo para reducir la longitud de la sesión. Es decir, hasta qué punto ayudan a los usuarios a encontrar la información que necesitan en el menor tiempo posible. Además, permite evaluar diferentes aspectos del algoritmo, incluyendo su precisión según el significado tradicional de dicho término.

Finalmente, nos hemos encontrado con varias limitaciones que deberían ser tratadas en futuros trabajos. En primer lugar, el uso de tres niveles de valoraciones (positiva, negativa, neutral) no funciona tan bien como habíamos esperado, principalmente debido a que la mayor parte de las consultas son, de hecho, neutrales según este esquema. Esto da lugar a recomendaciones incorrectas que, lo que es peor, son consideradas como buenas por las métricas de precisión en la predicción. Además, la idea de interpretar un *click* como éxito debería ser mejorada teniendo en cuenta más aspectos disponibles en el *query log*, como, por ejemplo, si el usuario ha visitado páginas de resultados adicionales, el tiempo transcurrido hasta que prueba con otra consulta, etc.

De la misma manera, en la realidad una única sesión puede contener consultas pertenecientes a varias búsquedas. Técnicas para superar este problema, segmentando el *query log* en búsquedas en lugar de sesiones, deberían ser también estudiadas.

Finalmente, sería interesante comparar los resultados de algoritmos de filtrado colaborativo con técnicas de *query suggestion* tradicionales, así como el desarrollo de nuevos algoritmos especialmente diseñados para este nuevo problema.

Referencias

[1] R. Baeza-Yates, C. Hurtado, and M. Mendoza, *Query Recommendation Using Query Logs in Search Engines*, vol. 3268/2004 of *Lecture Notes in Computer Science*, pp. 588–596. Springer Berlin / Heidelberg, November 2004.

[2] S. Cucerzan and R. W. White, "Query suggestion based on user landing pages," in *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, (New York, NY, USA), pp. 875–876, ACM Press, 2007.

[3] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating "word of mouth"," in *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, (New York, NY, USA), pp. 210–217, ACM Press/Addison-Wesley Publishing Co., 1995.

[4] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl, "Application of dimensionality reduction in recommender systems - a case study," in *In ACM WebKDD Workshop*, 2000.

[5] J. Wang, A. P. de Vries, and M. J. T. Reinders, "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," in *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, (New York, NY, USA), pp. 501–508, ACM, 2006.

[6] L. H. Ungar, D. P. Foster, E. Andre, S. Wars, F. S. Wars, D. S. Wars, and J. H. Whispers, "Clustering methods for collaborative filtering," AAAI Press, 1998.

[7] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *WWW '01: Proceedings of the 10th international conference on World Wide Web*, (New York, NY, USA), pp. 285–295, ACM, 2001.

[8] G. Pass, A. Chowdhury, and C. Torgeson, "A picture of search," in *InfoScale '06: Proceedings of the 1st international conference on Scalable information systems*, (New York, NY, USA), p. 1, ACM, 2006.

[9] "Removed for double blind submission,"

[10] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.

[11] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, pp. 43–52, 1998.

[12] J. Herlocker, J. A. Konstan, and J. Riedl, "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms," *Inf. Retr.*, vol. 5, no. 4, pp. 287–310, 2002.

[13] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: An open architecture for collaborative filtering of netnews," pp. 175–186, ACM Press, 1994.

[14] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, (New York, NY, USA), pp. 230–237, ACM, 1999.

[15] D. Lemire and A. Maclachlan, "Slope one predictors for online rating-based collaborative filtering," in *Proceedings of SIAM Data Mining (SDM'05)*, 2005.

[16] D. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles, "Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach," in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, UAI 2000*, (Stanford, CA), pp. 473–480, 2000.

[17] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *WWW '05: Proceedings of the 14th international conference on World Wide Web*, (New York, NY, USA), pp. 22–32, ACM, 2005.

[18] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, (New York, NY, USA), pp. 253–260, ACM, 2002.

RETOS EN EL DISEÑO DE SISTEMAS TELEMÁTICOS PARA LA COLABORACIÓN CIUDADANA

I. Seoane, D. Larrabeiti, A. García

Departamento Ingeniería Telemática

Universidad Carlos III de Madrid

Avda. Universidad 30, 28911 Leganés

{iseoane, dlarra, alberto} @it.uc3m.es

E. de la Hoz, M. A. López

Área de Ingeniería Telemática. Departamento de Automática

Universidad de Alcalá

Edificio Politécnico, Ctra. N-II, Km 36.500, 28871.Alcalá de Henares

{enrique.delahoz, miguelangel.lopez}@uah.es

J. Martinez

Grupo de Interconexion de Redes de Banda Ancha (GIRBA) Departamento de Comunicaciones Universidad Politécnica de Valencia

Camino de Vera, s/n 46022 Valencia

jmartinez@upvnet.upv.es

Resumen- El uso de redes Ad-Hoc inalámbricas para el soporte de situaciones de emergencia ha sido ampliamente estudiado en la literatura y en un gran número de proyectos. En la mayoría de sistemas, el flujo de información de colaboración ciudadana ha sido tradicionalmente gestionado de forma centralizada, a través de portales web, correo electrónico o telefónicamente, y se ha dedicado escaso interés a la colaboración ciudadana en modo ad-hoc. En este artículo analizamos las posibilidades de incorporar al ciudadano como un elemento participante y colaborativo en el reenvío de información y en el tratamiento cooperativo de situaciones especiales. Este paradigma presenta aplicaciones de interés, especialmente en el contexto de redes vehiculares, y retos tecnológicos interesantes, como el encaminamiento multicamino, el control de admisión, y la interacción de agentes aplicada a la colaboración ciudadana mútua, así como al descubrimiento de recursos de apoyo y la coordinación con las entidades oficiales de atención a una situación crítica.

Palabras Clave: tecnologías inalámbricas, situaciones de emergencia, redes ad-hoc, colaboración ciudadana, redes vehiculares, Multipath routing, MDC.

I. INTRODUCCIÓN

Entre las tecnologías de comunicaciones para la sociedad de la información, la movilidad de terminales con capacidad de transmisión de datos es un servicio clave para el tratamiento de emergencias, que se encuentra cada vez más presente en el equipamiento de personas y vehículos. Esta movilidad además ha surgido en paralelo con la reducción del tamaño de los dispositivos, sin que esto supusiera sacrificio de potencia de cómputo, almacenamiento y de capacidad de comunicación. Estos equipos portátiles, algunos de bolsillo, disponen de capacidad de autolocalización y proceso multimedia, y dispondrán en breve de más interfaces y tecnologías de comunicación inalámbrica (wifi, bluetooth, 3G, wimax, car2car), lo que aumenta la diversidad de posibilidades de interacción con el entorno.

Parece sensato pensar que toda esta potencialidad de cómputo y colaboración distribuida debe ser aprovechada en la atención a situaciones críticas, especialmente en aquellas que no disponen conectividad con la red fija debido a falta de cobertura, en muchas ocasiones debida a la propia naturaleza de la emergencia, como es el caso de una acción terrorista. En estas situaciones de crisis, tales como una catástrofe, un accidente o un acto criminal, parece realista proponer y justificar a los ciudadanos el uso abierto de recursos de computación y comunicación móviles y fijos privados en aras del bien común, garantizando la seguridad y el anonimato tanto de las comunicaciones como de los usuarios.

Existe un importante número de proyectos de investigación, por ejemplo [1][2][3][4][5][6], cuyo trabajo se centra en dotar a los distintos agentes y profesionales que intervienen en la gestión de crisis, de toda esta tecnología avanzada. Sin embargo, si bien la comunicación ad-hoc se ha planteado como una tecnología válida para las redes de emergencia, la mayoría de sistemas telemáticos de gestión de crisis incorpora la colaboración ciudadana mediante esquemas centralizados: centros de atención de emergencia telefónica, mensajería electrónica, aplicaciones web de alertas, foros de intercambio de ayuda, etc. La colaboración ad-hoc ha sido generalmente soslayada en estos sistemas, aunque, curiosamente, una de las formas más exitosas de colaboración ciudadana en el tratamiento de emergencias de la historia tiene naturaleza ad-hoc: la red de radioaficionados.

En este artículo analizamos algunas de las posibilidades de este nuevo paradigma de colaboración ad-hoc exploradas en el recientemente iniciado proyecto MCINN T2C2 [7] y describimos algunos retos tecnológicos implicados en la incorporación de los ciudadanos y sus dispositivos privados como elementos activos en la gestión distribuida de crisis. Las tecnologías emergentes de apoyo postuladas en este artículo, son las redes multiservicio con soporte al acceso multi-homed inalámbrico en movilidad, las redes vehiculares con enrutamiento geográfico, los sistemas inteligentes y el

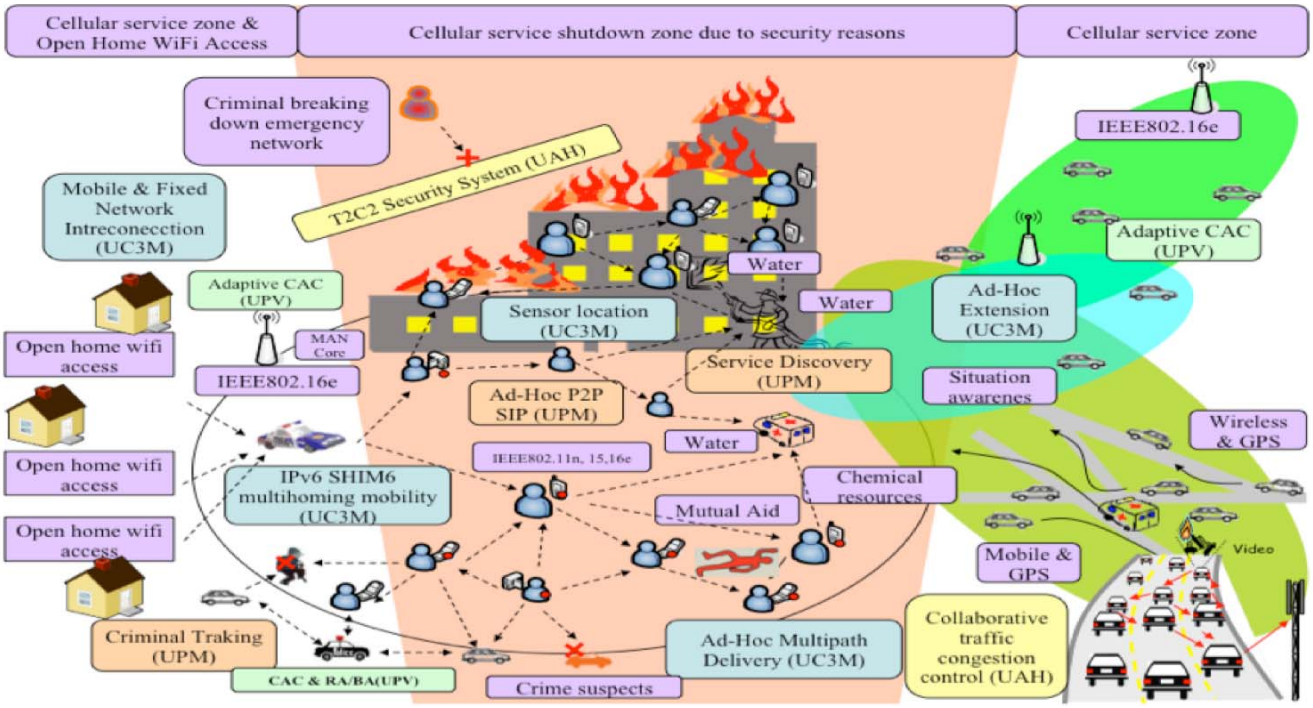


Fig. 1.- Escenario y tecnologías implicadas en gestión de crisis apoyada en la colaboración ciudadana

descubrimiento de servicios, nuevas arquitecturas de seguridad en sistemas aislados de una tercera parte confiable, mecanismos telemáticos de negociación, la interconexión de subredes ad-hoc con redes celulares de nueva generación, y su control de admisión en situaciones de emergencia.

Nos centramos en exponer los retos en dos aspectos concretos del escenario presentado: la cooperación ciudadana en redes vehiculares y el uso del multi-trayecto en una red ad-hoc con múltiples interfaces radio para aprovechar su capacidad de manera agregada en el transporte de flujos multimedia. Adicionalmente se analizarán los retos en el control de admisión en redes móviles multimedia para dar conectividad a red fija a los sistemas de emergencia con QoS.

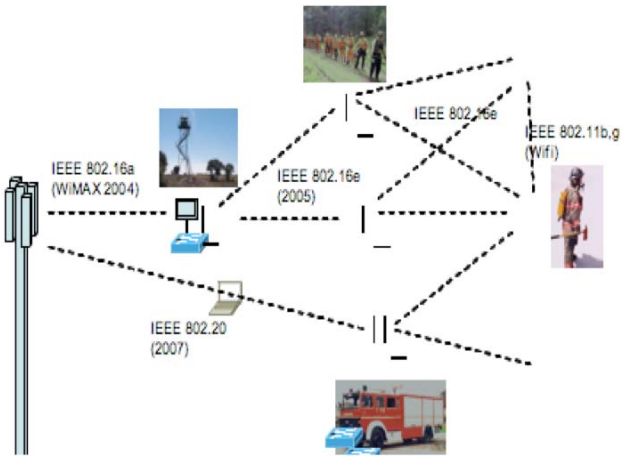


Fig. 2.- Redes ad-hoc extendiendo la cobertura de los sistemas celulares en el tratamiento de un incendio en el medio rural

También es importante prestar atención a las aplicaciones de colaboración ciudadana telemática en modo ad-hoc fuera de situaciones de crisis. Es necesario garantizar las prestaciones y la confidencialidad de los terminales, trabajando de forma anónima y segura en un segundo plano, aprovechando las capacidades desaprovechadas de los dispositivos. Algunos ejemplos simples de colaboración ciudadana donde se pueden aplicar los retos descritos son la cesión de la habilidad de captación y reenvío de información de su entorno desde dispositivos integrados en vehículos (cámara, posicionamiento, seguimiento de objetos) y PDAs particulares, el desarrollo de normas para crear de canales de emergencia en los puntos de acceso WiFi privados para dar soporte de movilidad a las unidades de emergencia, la localización y seguimiento telemático de sospechosos (por ejemplo, la identificación de coches y su seguimiento a través de VANETs), la colaboración inalámbrica entre navegadores GPS para organizar el tráfico rodado, etc.

II. Cooperación ciudadana en redes vehiculares

La gestión de los servicios de emergencia soportados en redes ciudadanas cooperativas da lugar a la aparición de escenarios enormemente complejos donde los aspectos fundamentales son la movilidad y la descentralización. Un reto emergente es la investigación y el desarrollo de tecnologías cooperativas en infraestructuras de servicios altamente dinámicos y distribuidos. Este desafío puede ser abordado desde la perspectiva de los sistemas complejos de cara a modelar y definir mecanismos de control de dichas infraestructuras. A este fin, proponemos el empleo de sistemas multiagente, en concreto de técnicas de negociación cooperativa. Estas técnicas se presentan como alternativas eficientes y prometedoras para aquellos escenarios donde los sistemas de control clásico fracasan debido a la dimensión, dinamismo y el carácter distribuido de los escenarios a controlar. Una aplicación sería por ejemplo la definición de

un servicio de gestión inteligente distribuido de tráfico rodado aplicado a situaciones de emergencia.

De manera transversal, las redes ciudadanas cooperativas presentan una casuística relativa a la seguridad similar a la existente en redes ad-hoc. Debido a que las redes cooperativas ciudadanas pueden también establecer comunicaciones con infraestructura de red fija, se propone una arquitectura para redes cooperativas ciudadanas que aprovechan las ventajas derivadas de disponer de una parte de infraestructura fija, pero también afrontando las amenazas potenciales que puedan provenir de la aparición dinámica de nodos colaboradores desconocidos. La solución de seguridad debe centrarse en el desarrollo de mecanismos para el establecimiento de confianza entre nodos y en la detección de nodos maliciosos o egoístas.

A. Desarrollo de tecnologías cooperativas para infraestructuras de servicios altamente dinámicas y descentralizadas

Debido al carácter dinámico y descentralizado de los escenarios de provisión de servicios en redes cooperativas, los sistemas multiagente se presentan como una solución adecuada que surge de manera natural para el modelado de sistemas complejos de cara a solventar los distintos problemas que surgen en las capas de aplicación y comunicaciones [8]. Los sistemas multiagente proporcionan una técnica denominada negociación cooperativa [9] que puede ser efectiva para afrontar esta tarea. La negociación cooperativa tiene como objetivo la resolución de problemas de optimización dinámica y distribuida multiobjetivo, donde un conjunto de entidades con intereses en conflicto interactúan dentro de un entorno dinámico. Esta técnica se ha aplicado con éxito en el campo del control aéreo, en gestión de red en redes ad-hoc o en problemas de asignación de recursos [10][11]. La negociación cooperativa es efectiva porque es robusta frente a la composición de la red de subsistemas que puede cambiar dinámicamente. Sin embargo, existen algunas cuestiones cruciales que aún están abiertas como la estabilidad de la negociación, su optimalidad [10], la convergencia en tiempo real [9], y su adaptación tanto a diferentes aplicaciones como a las condiciones cambiantes de una aplicación dada. La esencia de esta aproximación es sacar partido de los mecanismos competitivos para alcanzar acuerdos cooperativos. Específicamente, y tal y como sucede en escenarios de negociación competitiva, los agentes no comparten todo su conocimiento y sus interacciones se ajustan a un protocolo. Sin embargo y a diferencia de dichos entornos, los agentes no tratan de maximizar su utilidad desde una perspectiva egoísta sino que tratan de alcanzar acuerdos cooperativos eficientes. En lo referente a estos acuerdos, los agentes buscan soluciones como las descritas por la teoría de juegos cooperativa (solución Nash) y no por la teoría de juegos competitiva (equilibrio Nash). Los principales aspectos a abordar en el estudio de este tipo de negociaciones están centrados en el diseño de protocolos y estrategias de negociación de cara a obtener soluciones óptimas y robustas con respecto a la composición del sistema.

En este sentido, la noción más apropiada de la solución de un problema multiobjetivo es la pareto-optimalidad. Sin embargo, dado que habitualmente hay múltiples soluciones pareto-óptimas, es necesario discriminar entres ellas para obtener la más satisfactoria. Algunos de los criterios para tomar esta decisión son: la solución Nash, la solución de Kalai-Smorodinsky, la solución de Kalai o la maximización de la suma ponderada de las funciones objetivo. Junto con estos criterios de optimalidad es necesario considerar otros aspectos como estabilidad (¿consigue la negociación alcanzar siempre un acuerdo?), la convergencia en tiempo real (¿consiguen los agentes converger a un acuerdo dentro de un determinado plazo) y la adaptación a los cambios en el contexto (e.g.,¿pueden los mecanismos de negociación adaptarse a modificaciones del ancho de banda?)

B. Redes Vehiculares

Las redes vehiculares, también conocidas como VANET, son un ejemplo paradigmático de infraestructuras distribuidas y altamente dinámicas donde es posible desplegar servicios cooperativos de especial interés. Ejemplos posibles son la gestión de tráfico en emergencias y el reconocimiento cooperativo de matrículas de vehículos. En la actualidad, existe un esfuerzo significativo de investigación en redes de vehículos por parte de grupos de investigación europeos y consorcios como Car2Car y en estándares en estudio como WAVE IEEE 802.11p que han alcanzado una madurez reseñable en la definición arquitectural, aunque algunos aspectos aún necesiten una definición más clara. En el marco de este proyecto, los vehículos actúan como nodos cooperantes en escenarios de emergencias. Estos escenarios plantean dos desafíos: garantizar la conectividad entre los nodos cooperantes y establecer los servicios de colaboración. Con respecto al primero, merece la pena señalar que los protocolos para las redes ad-hoc móviles comienzan a alcanzar cierta madurez, en particular en redes vehiculares [12][13]. El problema del direccionamiento, sin embargo, parece especialmente relevante. En todas las propuestas para encaminamiento geográfico-jerárquico, el direccionamiento geográfico aparece como parte de la capa de encaminamiento. Cada propuesta emplea un direccionamiento específico, lo que afecta a la compatibilidad entre aplicaciones y fuerza a adaptaciones adicionales. El encaminamiento a través de direccionamiento simbólico de mensajes jerárquico geográfico [14] parece asimismo prometedor. Parece factible mejorar los protocolos existentes o desarrollar algunos nuevos del tipo jerárquico geográfico. En el área de servicios colaborativos en redes vehiculares, el principal reto encontrado es desarrollar técnicas de negociación cooperativa, por ejemplo, soluciones para la gestión de tráfico en escenarios de emergencia. Es importante resaltar que aunque existen importantes precedentes en el uso de sistemas multiagente para la gestión de tráfico [8][16] y estas soluciones modelan el problema de forma distribuida, la toma de decisiones es fundamentalmente centralizada.

C. Seguridad en redes ad-hoc

Las redes ad-hoc conllevan nuevos riesgos de seguridad que o bien no son afrontados por las redes clásicas, o no pueden ser resueltas mediante la aplicación de aproximaciones convencionales. Las principales diferencias residen en las condiciones del escenario: acceso inalámbrico, nodos con recursos limitados y topología dinámica con movilidad de nodos continua y no predecible, que hace inútil

emplear soluciones centralizadas (autoridades de certificación, servidores de distribución de claves, servidores de autorización…). Esto ha conducido a la búsqueda de nuevas soluciones de seguridad ligeras basadas en mecanismos distribuidos. Un aspecto clave para garantizar la supervivencia de una red ad-hoc es la seguridad de los protocolos de encaminamient. Por ello, la mayor parte del esfuerzo investigador se ha centrado en este aspecto [17]. Debido a la naturaleza de los nodos, se hace necesario el empleo de mecanismos criptográficos ligeros para poder ofrecer servicios de seguridad sobre la información sensible intercambiada por los nodos.

Una diferencia importante de las redes vehiculares con respecto a las redes ad-hoc puras, es que es posible asumir la existencia de cierta infraestructura fija asociada a las propias vías por las que discurre el tráfico. Por tanto, se puede emplear parte de esta infraestructura para apoyar los mecanismos de establecimiento de confianza y autenticación entre los nodos de la red. Esta infraestructura podría servir, por ejemplo, para la distribución de certificados a los nodos. En un escenario de este tipo que emplee certificados, el acceso a información de revocación de los mismos es un elemento crítico. Por tanto, es necesario explorar mecanismos progresivos que permitan que el sistema se mueva desde mecanismos puros de acceso hasta información como OCSP a otros puros de una red ad-hoc en función de la situación e historia pasada del nodo. La implementación de mecanismos de evaluación de la confianza permitiría asignar un valor de confianza a cada nodo. Finalmente, aparece otra línea interesante, como es la adaptación y el desarrollo de nuevos protocolos para la protección de las comunicaciones entre los nodos adaptadas a este escenario.

III. Soporte Multi-Trayecto en una Red Ad-Hoc

Dentro del paradigma de la colaboración ciudadana y de la atención a emergencias, la solución a los retos anteriormente comentados pasaría por uno más: el diseño de una arquitectura de protocolos que permita explotar la red en modo ad-hoc de la manera más eficiente posible. Teniendo en cuenta las necesidad de las aplicaciones y servicios que pueden aplicarse a estos escenarios, los principales retos se concentran en: la movilidad de la red y su interconexión segura con la infraestructura troncal fija disponible en el entorno mediante el uso de nuevos protocolos de red con multi-homing transparente a transporte/aplicación [19], las prestaciones de la red al interconectar tecnologías de acceso inalámbrico distintas mediante nodos con múltiples interfaces, la distribución de contenido multimedia en tiempo real sobre la red ad-hoc, el aprovechamiento de la diversidad de caminos extremo a extremo, el despliegue de redes de sensores interconectadas a redes ad-hoc inalámbricas para la medición de datos en el entorno de emergencia, etc.

La red carecerá generalmente de infraestructura, pero puede aprovechar la existencia de otras redes, tanto fijas como inalámbricas en las cercanías, de forma que se pueda acceder en algún momento a una red troncal.

A. Redes Ad-Hoc Multicamino

De entre todas las posibles aplicaciones que puedan proponerse sobre esta arquitectura de red, las más exigentes, en cuanto a prestaciones y calidad serán aquellas que involucren la transmisión multimedia en tiempo real. Es por ello que el interés de la investigación en multimedia ad-hoc se ha centrado en obtener una distribución de la mayor calidad posible con las limitaciones impuestas por una red muy cambiante: ancho de banda limitado, retardos muy variables en función de la densidad de nodos, alto nivel de interferencias en entornos inalámbricos densamente poblados (los entornos WiFi a la frecuencia de 2.4GHz están habitualmente muy saturados en zonas urbanas por la rápida acogida de esta tecnología para fines personales) que provocan errores en la transmisión, capacidad de almacenamiento reducida y duración de la batería limitada.

Con estos problemas intrínsecos al escenario objetivo, parece correcto pensar en el uso de técnicas de distribución de datos avanzadas que puedan explotar una característica común en estas redes autoorganizadas: la existencia de varios caminos entre el origen y el destino y la posibilidad de transmitir datos desde diferentes orígenes a un mismo destino [20][21].

B. Distribución de múltiples descripciones de video

La distribución de video desde uno o múltiples orígenes a un mismo destino aprovechando múltiples caminos no interferentes de la red, permite por un lado dotar de robustez a la transmisión, dado que reduce la probabilidad de pérdida de información en caso de caída de un camino, y hacer un uso más amplio del ancho de banda agregado disponible. Tradicionalmente, los sistemas que pretendían evitar la pérdida de datos en estos escenarios hacían uso de técnicas de FEC (o ARQ para tráfico elástico). La redundancia puede enviarse empleando la multiplicidad de caminos. En el caso concreto del vídeo, existe una solución complementaria todavía escasamente probada en la práctica que es el uso de técnicas de codificación multidescripción (MDC) [22][23], apoyada en algoritmos de encaminamiento y reenvío multicamino o multitrayecto apropiados.

Las técnicas FEC sufren el inconveniente de añadir datos redundantes de forma proporcional a la probabilidad de error, realizándose el diseño en base a un peor caso. Esto supone que la ganancia de eficiencia en el uso de ancho de banda disminuye en situaciones donde la probabilidad de error sea alta. Además, en situaciones donde la calidad de los enlaces sea muy variable, puede suceder que una vez sacrificado cierto nivel de redundancia para la protección, la probabilidad de que los datos lleguen sin error sea similar a la de las ocasiones en las que los errores superen la capacidad de corrección del código elegido, con lo que en ambos casos, los datos redundantes son desperdiciados. Esto lleva al diseño de técnicas de FEC adaptativo que tienen los problemas de tiempo de reacción estudiados en teoría de control, afectando en estos intervalos a la calidad de la experiencia (QoE) percibida. Para mejorar la eficiencia de los códigos FEC se usan además técnicas de entrelazado, pero éstas provocan otro efecto que es el aumento en la latencia de las transmisiones debido al retardo de decodificación del entrelazado.

Las técnicas de ARQ tampoco son deseables para los escenarios propuestos, ya que no podemos asegurar con una alta probabilidad la existencia de un bucle cerrado que permita el control sobre las retransmisiones y su latencia puede no resultar aceptable.

En cualquier caso, estas técnicas suelen estar habitualmente implementadas con un alto grado de efectividad en los dispositivos físicos de los equipos para dotar de cierta robustez a los niveles inferiores de la torre de protocolos. Es por ello que en lugar de proponer cambios en los algoritmos implementados por los fabricantes en niveles 1 y 2, se propone el uso alternativo de codificación de los datos y de la manera de encaminarlos y reenviarlos por la red hacia el destino, por caminos disjuntos.

Las técnicas de distribución de video mediante codificación con múltiples descripciones, permiten el envío de flujos independientes con requisitos de prestaciones de red similares, que en caso de ser recibidos de manera parcial, permiten al receptor decodificar el video con una calidad mínima aceptable a partir de las descripciones recibidas. La mejora se obtiene por la combinación de las descripciones, aprovechando la información no redundante que lleva cada una de las recibidas.

A diferencia de otras técnicas de codificación (como por ejemplo Layered Coding- LC) [24], no existe ninguna jerarquía entre las descripciones, dotando al sistema de una mayor flexibilidad, dado que no hay que proteger más ningún flujo frente a los demás.

C. Encaminamiento Multicamino y MDC

Para poder usar de forma eficiente la codificación MDC tiene que darse la posibilidad de usar más de un camino simultáneamente entre un mismo origen y un destino, o disponer de más de una fuente sirviendo el mismo contenido con un receptor común. En el primer caso, el diseño de la arquitectura de la red debe de incorporar algoritmos de encaminamiento que permitan en primer lugar el descubrimiento de más de una ruta posible, y en segundo lugar, que incorporen mecanismo de reenvío a nivel de red o de aplicación que utilicen esas rutas descubiertas simultáneamente.

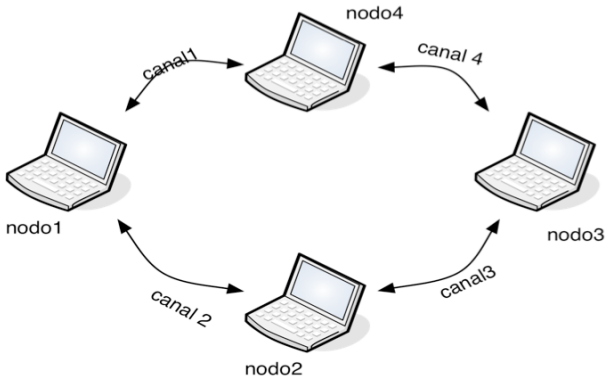


Fig. 3 - Esquema de la red ad-hoc de pruebas

Se puede demostrar de manera sencilla las posibilidades de esta técnica modificando prototipos de protocolos de enrutamiento disponibles en la comunidad investigadora. Por ejemplo, existen modificaciones de la implementación del protocolo de routing AODV [25] para Linux para descubrir caminos disjuntos dentro de la red ad-hoc y mantener la información actualizada a petición de los nodos de origen de las aplicaciones [26]. Si ampliamos esta capacidad con una facilidad de planificación de canales dinámica distribuida se puede montar una red ad-hoc evitando tecnologías y canales interferentes entre sí [27][28]. En la Fig. 3 puede verse un sencillo esquema de una maqueta de pruebas en los que existen cuatro nodos conectados en ad-hoc. Cada uno de ellos dispone de dos interfaces de red, con lo que podemos asignar 4 canales no interferentes distintos entre ellos, en función de la calidad de la señal que recibe cada nodo.

Una vez decididas las frecuencias de trabajo y la configuración de la red, el algoritmo de encaminamiento multicamino basado en AODV permite descubrir caminos disjuntos sobre la red bajo demanda del nodo que incia el envío de datos. A diferencia de otras, esta implementación proporciona información para diseñar aplicaciones que usen estos caminos simultáneamente, y no sólo como medida de protección o backup.

A su vez, las aplicaciones de atención y colaboración con el ciudadano serán las encargadas de negociar las sesiones de red de forma que puedan sacar el máximo partido a las prestaciones del escenario multicamino: negociación de la carga de datos multimedia, descubrimiento de fuentes con descripciones correladas para aumentar la calidad, etc.

El reto y su aplicación se ilustran en una sencilla red (Fig. 3). El siguiente experimento de transmisión de video entre los nodos 1 y 3 de la figura (usando la aplicación VLC y un video MPEG2 codificado a un bitrate variable máximo de 6Mbps) ilustra con este escenario el comportamiento de la aplicación cuando puede aprovechar la existencia de dos caminos. En el primer caso (Fig. 4), el protocolo de encaminamiento sólo encuentra un camino posible (caso de caída de un enlace), mientras que en el segundo caso (Fig 5), el algoritmo AODV-Modificado permite encontrar dos rutas posibles.

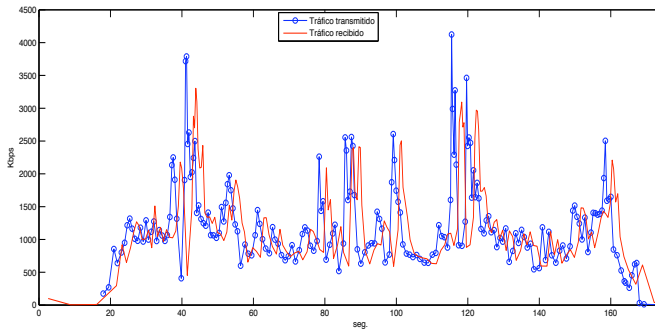


Fig. 4.- Transmisión de un video por la red Ad-Hoc inalámbrica en bps usando un camino con un salto (tramas enviadas –o–, tramas recibidas sin error —)

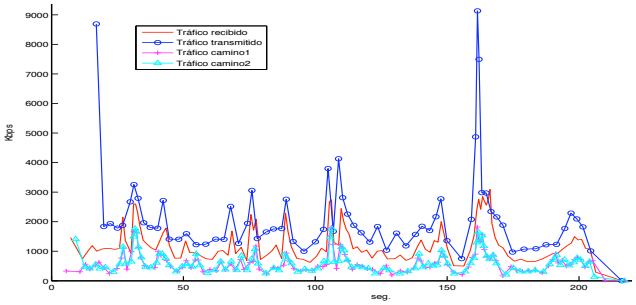


Fig. 5- Tráfico generado por la transmisión de un video por la red Ad-Hoc inalámbrica en bps usando 2 caminos (camino1 –^–, camino2 –+–, tramas enviadas –o–, tramas recibidas sin error —).

Para simplificar el reenvío, se ha usado un sencillo sistema en el que las descripciones están compuestas por el 50% de los paquetes del flujo completo (balanceando ambos enlaces con la misma carga).

Como puede observarse, el uso del multipath permite la recepción de un bitrate agregado mayor, consiguiéndose una calidad perceptual mayor en la visualización del video enviado reduciendo el tráfico en cada enlaces y reduciendo el retardo medio que sufren los paquetes en los caminos con más de un salto.

Situaciones donde la aplicación tenga necesidades de ancho de banda mayores presentan otra problemática relacionada con el retardo en los caminos debido al número de saltos. La demostración de la relación entre la calidad perceptual y las distribuciones de retardo que sufre el tráfico multimedia en estos escenarios son parte de los trabajos en curso actuales

En el caso de que dispongamos de varias fuentes de datos sirviendo diferentes descripciones del mismo contenido, podemos utilizar "overlays" en las que los receptores solicitan los datos mediante protocolos P2P [22][29] o también puede usarse aplicaciones multicast, tanto a nivel de red como a nivel de aplicación (ALM) donde el receptor conoce los grupos a los que subscribirse en el caso de poder o querer recibir más de una descripción.

D. Conclusiones

Como conclusión, cabe destacar en los resultados el correcto funcionamiento del algoritmo de encaminamiento ad-hoc modificado para poder simultanear el envío por todas las interfaces posibles en un dispositivo, de forma que aprovechemos la existencia de caminos disjuntos origen-destino.

Es interesante plantear la comprobación de la mejorar la eficiencia de los algoritmos implementados, de forma que no sólo podamos asegurar un cierto grado de robustez en cuanto a la calidad percibida, sino también en la reducción de la latencia. Esta disminución del tiempo de reproducción, permitiría el uso de las técnicas de distribución de video con MDC-Multipath en aplicaciones donde la rapidez de la recepción sea crítica, como es el caso de los entornos de colaboración ciudadana descritos en la introducción. Debido a la heterogeneidad de la red, la distribución de probabilidad de retardo en cada uno de los posibles caminos varía mucho, ya que no sólo puede haber involucrados más de un dispositivo fuente [29], sino también distintos tipos de tecnologías de comunicación. Esto supone que las distintas descripciones se van a recibir con una latencia distinta, y que los paquetes de cada una de ellas sufrirán retardos diferente al del resto de descripciones.

Por tanto, en función de los requisitos de la aplicación colaborativa que queramos desplegar, existirá una relación de compromiso entre la calidad percibida y la rapidez a la que se quiere lograr recibir los datos con ese nivel de calidad. Esto es mucho más flexible en el momento en que se disponga de varias descripciones, pudiendo dar un servicio más rápido pero adecuado a la calidad menor obtenida de decodificar los paquetes más rápidos, pero asegurando un nivel continuado mayor que en el caso de una sola descripción.

En paralelo con éstos, también hay que tener en cuenta la necesidad de dotar a la red cooperativa de movilidad entre infraestructuras fijas y móviles, por ejemplo usando algoritmos basados en SHIM6.

IV. Control de Admisión en Redes Móviles con Flujos Adaptativos en Tasa y Arquitectura Jerárquica

El éxito esperado de tecnologías inalámbricas como WiMAX, o la nueva MobileFi, permitirán construir redes superpuestas de bajo coste, tanto sobre redes de acceso 3G o WiFi. Estas soluciones encontrarán su aplicación en nuevos escenarios como el despliegue de redes de emergencia robustas. La gestión de eficiente de recursos es un requerimiento fundamental en estas redes, en las que la carga ofrecida puede ser difícil de estimar y por tanto los recursos pueden no estar dimensionados adecuadamente.

Es de esperar que en las redes de emergencia coexistan tanto trafico streaming (se considera tanto el tráfico de tiempo real pregrabado como interactivo) como elástico. Con la llegada de nuevas técnicas de codificación, la tasa de transmisión de una gran porción de aplicaciones de tiempo real puede ser adaptada típicamente entre un conjunto discreto de valores. En este escenario, es crucial el diseño de nuevas políticas integradas de control de admisión y adaptación de tasa que se adapten a condiciones cambiantes de tráfico y exploten información predictiva sobre la futura ocurrencia de traspasos.

Las nuevas políticas permitirán garantizar la calidad de servicio (QoS) definida en términos de cotas para la probabilidad de bloqueo de nuevas sesiones así como para la probabilidad de terminación forzosa de sesiones en curso y de esta forma maximizar el tráfico cursado por el sistema. Estas políticas deben también minimizar las perturbaciones observas por los usuarios de aplicaciones streaming debidas a adaptaciones de tasa inadecuadas, así como limitar la probabilidad de abandono de los flujos elásticos debida a que su tasa disponible esté por debajo de un mínimo. Una aproximación al control de admisión que considere los objetivos de bloqueo, terminación forzosa, abandono de flujos elásticos y adaptación de tasa de forma integrada permite mejorar considerablemente la eficiencia del sistema.

Por otra parte, el despliegue de redes móviles superpuestas con estructura jerárquica hará necesario el diseño de herramientas de planificación que sean computacionalmente eficientes y que permitan estimar la capacidad del nuevo sistema. Así mismo, será necesario el diseño de esquemas de control de admisión que gestionen de forma eficiente el tráfico de desbordamiento.

V. Conclusiones

En este artículo se han analizado las posibilidades de incorporar al ciudadano como un elemento participante y colaborativo en el reenvío de información y en el tratamiento cooperativo de situaciones especiales. Este paradigma presenta retos tecnológicos importantes, de los cuales nos hemos centrado especialmente en tres: la cooperación entre sistemas de navegación a bordo en vehículos, el uso efectivo de multiplicidad de trayectos para obtener una mayor utilización de la red en la transmisión de flujos multimedia y el control de admisión en redes 4G con flujos adaptativos.

AGRADECIMIENTOS

Los resultados de este trabajo se engloban dentro del proyecto TIN2008-06739-C04/TSI.

REFERENCIAS

[1] ISO/TC 223 Group. http://www.iso.org/iso/standards_development/technical_committees/other_bodies/iso_technical_committee.htm?commid=295786

[2] ORCHESTRA. http://www.eu-orchestra.org/

[3] OASIS. http://www.oasis-open.org/

[4] AKOGRIMO. http://www.akogrimo.org/

[5] U2010. http://www.u-2010.eu/

[6] IMPROVISA. http://www.gsi.dit.upm.es/~improvisa/

[7] T2C2. http://adscom01.it.uc3m.es/~T2C2

[8] Gerhard Weiss. "Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence". MIT Press, Cambridge MA, USA, 1999

[9] John Bigham and Lin Du. "Cooperative negotiation in a multi-agent system for real-time load balancing of a mobile cellular network". In AAMAS '03: Proceedings of the second international jointconference on Autonomous agents and multiagent systems, pages 568–575, New York, NY, USA,2003. ACM.

[10] D.M.; Tomlin C.J. Inalhan, G.; Stipanovic. "Decentralized optimization, with application to multiple aircraft coordination". volume 1, pages 1147–1155, 2002.

[11] Peter Marbach and Ying Qiu. "Cooperation in wireless ad hoc networks: a market-based approach". IEEE/ACM Trans. Netw., 13(6):1325–1338, 2005.

[12] "Car to car consortium Manifiesto". http://www.car-2-car.org/fileadmin/dokumente/pdf/C2CCC_manifesto_v1.1.pdf.

[13] "IEEE 1609 - Family of Standards for Wireless Access in Vehicular Environments (WAVE)" [IEEE802.11] IEEE 802.11 Task group p. http://grouper.ieee.org/groups/802/11/Reports/tgp_update.htm

[14] Durr, F.; Becker, C.; Rothermel, K., "Efficient forwarding of symbolically addressed geocast messages,"Computer Communications and Networks, 2005. ICCCN 2005. Proceedings. 14th International Conference on , vol., no., pp. 77-83, 17-19 Oct. 2005

[15] A. Boukerch, L. Xu and K. EL-Khatib. "Trust-based security for wireless ad hoc and sensornetworks". Computer Communications. Special issue on security on wireless ad hoc and sensornetworks. Volume 30, issues 11-12. 10 September 2007, pp. 2413-2427.

[16] Ruimin Li; Qixin Shi, "Study on integration of urban traffic control and route guidance based on multi-agent technology," Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE , vol.2, no., pp. 1740-1744 vol.2, 12-15 Oct. 2003.

[17] Panagiotis Papadimitratos and Zygmunt J. Haas, "Secure routing for mobile ad hoc networks, in SCS Communication Networks and Distributed Systems". Modeling and Simulation Conference (CNDS) 2002. 15

[18] G. Ibáñez, A. Azcorra, A. García-Martínez. "Procedimiento de gestión de enlaces en el nivel de enlace de datos para redes de comunicaciones, procedimiento de encaminamiento de tramas para datos, dispositivo de interconexión de redes y red que combina ambos procedimientos". Patente de invención. Solicitud P200702358. Agosto 2007.

[19] E. Nordmark, M. Bagnulo. "SHIM6: Level 3 Multihoming SHIM Protocol for IPv6", drafietf-shim6-proto08, May2007.

[20] A. C. Begen, Y. Altubasak, O. Ergun, M. Ammar. „Multi-path selection for multiple description video streaming over overlay networks". Signal Processing Image Communications 20. 39-60. 2005.

[21] A.G. Apostolopoulos, M. D. Trott. "Path diversity for Enhaced Media Streaming". IEEE Communications Magazine, August 2004.

[22] A. Vitalli, M. Fumagalli. "Standard-Compatible Multiple-Description Coding (MDC) and Layered Coding (LC) of Audio/Video Streams". <draft-vitali- ietf-avt-mdc-lc-00.txt> CEFRIEL – Politecnico di Torino 2005.

[23] Manuela Pereira, Marc Antonini Michael Barlaud. "Multiple Image and Video Coding for Wireless Channels". Signal Processing Image Comunnications 18. 925-945. 2003.

[24] J.Shareski, S.Han, B.Girod. "Layered Coding vs. Multiple Descriptin for Video Streaming over Multiple Paths". Multimedia Systems, January 2005. Springer –Verlag.

[25] "Ad-Hoc On Demand Distance Vector (AODV) Routing Algorithm". Implementacion de la universidad de Upsala. http://core.it.uu.se/core/index.php/AODV-UU .

[26] Gerson Rodríguez de los Santos, Isaac Seoane, López, David Larrabeiti. "Soporte Multitrayecto en red ad-hoc basada en una propuesta de extensión de AODV". Actas XVIII Jornadas Telecom I+D-2008.ISBN-13: 978-84-9860-135-0.

[27] "Zero Configuration Networking". http://www.zeroconf.org/

[28] RFC 3927, "Dynamic Configuration of IPv4 Link-Local Addresses" S.Cheshire, B.Aboba, E. Guttman. May 2005

[29] W.P. Ken Yiu, X. Jin, S. H . Gary Chan. Challenges and Approaches in Large-Scale P2P Media Sttreaming. IEEE Multimedia Volume 14 nº2. 2007.

Estudio de prestaciones de los algoritmos de predicción LZ

Alicia Rodríguez-Carrión, Carlos García-Rubio, Celeste Campo
Departamento de Ingeniería Telemática
Universidad Carlos III de Madrid
Avda. Universidad, 30, Leganés (Madrid), Spain
arcarrio@it.uc3m.es, cgr@it.uc3m.es, celeste@it.uc3m.es

***Resumen*—El problema de la predicción de la siguiente localización se da en un gran número de situaciones. Aplicado a redes móviles, sabiendo la siguiente localización de un usuario se podrían gestionar los recursos de la red de forma más eficiente y ofrecer servicios de información relacionada con dicha posición futura. Este artículo presenta un estudio de la familia de algoritmos LZ para la predicción de la siguiente localización. Con estos algoritmos y la introducción de algunas modificaciones se han conseguido probabilidades de acierto de hasta un 90%. Además, se incluye un estudio de los recursos consumidos por cada algoritmo para evaluar si es posible su ejecución en terminales móviles.**

***Palabras Clave*—predicción, localización, algoritmos LZ**

I. Introducción

Según se comenta en el Análisis de la Sociedad de la Información en España 2008 [1], el número de líneas móviles sigue creciendo de forma significativa según la tendencia de años anteriores. Además, el estudio también pone de relieve el crecimiento que se está produciendo en los últimos tiempos en el acceso a Internet desde terminales y redes móviles. Sin embargo, esta evolución del número de usuarios y velocidades de acceso lleva asociados nuevos problemas y retos.

El crecimiento del número de suscriptores obliga a utilizar celdas cada vez más pequeñas, por lo que el número de usuarios que cruzan la frontera entre dos celdas consecutivas con una llamada en curso se ve notablemente incrementado. Cualquier terminal móvil que cruce la frontera entre dos celdas se verá obligado a realizar un traspaso de los recursos necesarios (*handover*). Para ello, algunos esquemas [2] reservan recursos en algunas o todas las celdas vecinas, de forma que tratan de asegurar cierta calidad de servicio, pero esto supone un uso poco eficiente de los recursos radio, que con el aumento de ancho de banda demandado por cada usuario son cada vez más escasos. Si se supiera la próxima localización del usuario móvil, dichos recursos se reservarían en la celda correspondiente a dicha localización futura, lo que reduciría la latencia de acceso y mejoraría el uso de los recursos.

Por otro lado, la reducción del tamaño de las celdas da lugar a unas áreas de localización (LA) más pequeñas, lo que conlleva un aumento del tráfico de señalización para tareas de gestión de la movilidad (*paging* y *location area updates*). Puesto que este tráfico se envía por el interfaz aire, cuello de botella de toda red móvil, interesa reducir su volumen. Esto se podría conseguir si se conociese cuál será la siguiente localización del usuario [3].

Existen diferentes formas de abordar el problema de la predicción de la localización, pero este artículo se centrará en aquella que usa los algoritmos de predicción de la localización. De todos los algoritmos de predicción existentes [4], el estudio se ocupará de la familia de algoritmos LZ puesto que sus características principales (predicciones en tiempo real y posibilidad de ejecución en un terminal móvil) son muy interesantes para los fines comentados anteriormente.

Los objetivos de este artículo son básicamente dos. En primer lugar se hará un breve comentario sobre la implementación de estos algoritmos, que dio como resultado una aplicación gracias a la cual se pueden procesar trazas de movilidad y analizar aspectos como la probabilidad de acierto de los algoritmos y los recursos que consumen. El segundo objetivo, y más importante, será precisamente este análisis de prestaciones y por lo tanto será el tema principal de este estudio.

El resto del artículo se estructura de la siguiente forma. La segunda sección trata de dar una noción básica del funcionamiento de los algoritmos de predicción de la familia LZ. En la sección III se comentan los aspectos de la implementación de los algoritmos y en la sección IV se exponen los resultados y análisis del procesado de trazas de movilidad de un usuario. Para finalizar, en la sección V se resumirán las conclusiones que se pueden extraer de los análisis realizados.

II. Algoritmos de predicción LZ

Los algoritmos de predicción LZ, diseñados por Ziv y Lempel [5], son algoritmos incrementales muy populares para realizar análisis de cadenas de texto. Se clasifican dentro de los algoritmos independientes del dominio, y por tanto consideran las localizaciones como símbolos, sin tener en cuenta otras semánticas de la localización (coordenadas, función del lugar, ...).

Puesto que la red que se va a tener en cuenta en el contexto de este artículo es la red celular, se considerará un sistema basado en celdas agrupadas en áreas de localización (LA). Cada celda se identificará mediante un número resultante de la concatenación del código de LA (LAC) y del identificador de celda (CellID), aunque por simplicidad en la presente descripción se utilizarán distintas letras para designar cada una de las celdas.

Los algoritmos de predicción obtienen la información necesaria del denominado **historial de movimientos**. Dicho historial es una cadena de símbolos, $L = a_1 a_2 \dots a_n$, que representan las celdas por las que el usuario ha transitado durante su desplazamiento. Para obtener esta cadena de símbolos el dispositivo móvil almacena en un archivo el LAC y CellID cada vez que el usuario cambia de celda, de

forma que la información de movilidad puede ser procesada posteriormente. Con este historial y partiendo de la suposición básica de que el comportamiento de los usuarios sigue un modelo probabilístico (proceso estocástico) y repetitivo (proceso estacionario), los algoritmos hacen su predicción en dos fases:

1) Construcción y actualización de la información almacenada (en forma de árbol) tras procesar el nuevo símbolo.
2) Cálculo de la probabilidad de que cada símbolo conocido sea el correspondiente a la siguiente localización.

La primera fase está basada en el modelo de Markov O(k) [4], con la excepción de que k es variable y puede crecer hasta infinito, mientras que para la segunda fase se pueden usar distintas técnicas. La ventaja de los algoritmos de esta familia es que funcionan de tal forma que el crecimiento de k es óptimo, es decir, si en un momento dado se tratara de usar un orden k mayor del que se está usando, los resultados no mejorarían. En algunos estudios [3] [6] se habla del uso de un indicador de la mejor predicción secuencial posible que se puede hacer sobre una secuencia arbitraria de símbolos de entrada, denominado previsibilidad de estados finitos. En dichos artículos se asegura que el modelo LZ supera asintóticamente un modelo de Markov de cualquier orden y alcanza dicha previsibilidad de estados finitos. Los tres algoritmos que se estudiarán en detalle en las siguientes secciones se distinguen por la rapidez con la que tratan de llegar a estos resultados óptimos mediante la variación de la construcción de los árboles asociados a cada uno de ellos y el uso de distintas técnicas de cálculo de probabilidades.

A. Algoritmo LZ

Este algoritmo es el más básico de los tres y el punto de partida de los dos restantes. Su funcionamiento es como sigue. Sea γ la cadena vacía y dada una cadena de entrada L, el algoritmo LZ divide dicha cadena en distintas subcadenas $s_0, s_1, \ldots, s_m$ tales que $\gamma = s_0$ y las cadenas s_j, quitando su último símbolo, son iguales que alguna cadena s_i, para $j \geq 0$ y $L = s_0 s_1 ... s_m$. Obsérvese que la división se hace secuencialmente, es decir, tras determinar cada s_i, el algoritmo sólo considera la cadena de entrada restante. Tomando como ejemplo la cadena $L = abababcdcbdcbacababcbabcab$, la división resultante sería la siguiente: γ, a, b, ab, abc, d, c, bd, cb, ac, aba, bc, ba, bca, b.

Existe un árbol, denominado árbol LZ, que crece de forma dinámica durante el proceso de análisis y división de la cadena. Cada nodo del árbol representa una subcadena s_i, y almacena además un contador de aparición de dicha subcadena en L. El árbol resultante del análisis y división de la cadena del ejemplo es el mostrado en la Figura 1. Una vez actualizado el árbol en cada nueva iteración, se necesita estimar la probabilidad de que cada símbolo conocido sea el siguiente en aparecer. Para ello, Vitter [7] sugiere utilizar la expresión 1:

$$P\left(X_{n+1} = a|L\right) = \frac{N^{LZ}\left(s_m a, L\right)}{N^{LZ}\left(s_m, L\right)} \qquad (1)$$

donde $N^{LZ}\left(s_m a, L\right)$ expresa la frecuencia del nodo $s_m a$ y $N^{LZ}\left(s_m, L\right)$ representa la frecuencia del nodo s_m en el árbol LZ. Para realizar la predicción, el algoritmo LZ escoge el

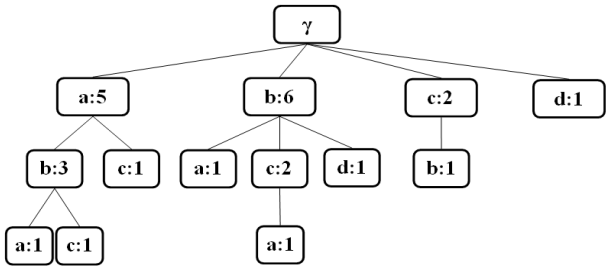


Fig. 1. Árbol LZ resultante de analizar la cadena del ejemplo

símbolo que tenga el estimador de mayor probabilidad, es decir, el mayor valor de $P\left(X_{n+1} = a|L\right)$.

Este algoritmo posee dos importantes inconvenientes:

- Toda la información referente a cadenas formadas por la unión de dos subcadenas consecutivas se pierde. En el ejemplo, se tiene $s_7 = bd$ y $s_8 = cb$, pero la cadena dc no se existe en el árbol LZ.
- Asimismo, tampoco se consideran las subcadenas dentro de los patrones s_i. En el ejemplo inicial, existe la cadena bca mientras que ca no aparece en el árbol LZ.

B. Algoritmo LeZi Update

Bhattacharya y Das [3] proponen una modificación en la construcción del árbol, así como una forma de usar dicho árbol modificado para predecir el siguiente símbolo, salvando el inconveniente de las subcadenas dentro de patrones s_i que no se tienen en cuenta.

Según esta modificación, cuando se crea una nueva hoja del árbol para el nodo s_i, todos los sufijos propios de s_i, es decir, todos los sufijos sin incluir s_i, también se insertan en el árbol. Todo esto se hace teniendo en cuenta que los nodos correspondientes a los sufijos propios de la cadena s_i insertados de esta forma no se tienen en cuenta a la hora de analizar y dividir la cadena de entrada restante, sino que este proceso se hará siguiendo estrictamente el algoritmo LZ y teniendo en cuenta únicamente los nodos que añadiría éste. El resto de nodos añadidos siguiendo las modificaciones introducidas por el algoritmo *LeZi Update* sólo se consideran para el cálculo de probabilidades, y se marcarán para poder diferenciarlos. Esto evita que el orden del modelo subyacente, k, alcance valores más altos de lo óptimo en cada iteración. De esta forma, la división de la cadena del ejemplo inicial será como se indica a continuación: γ, a, b, $ab\,\{b\}$, $abc\,\{bc, c\}$, d, c, $bd\,\{d\}$, $cb\,\{b\}$, $ac\,\{c\}$, $aba\,\{ba, a\}$, $bc\,\{c\}$, $ba\,\{a\}$, $bca\,\{ca, a\}$, b. Las cadenas entre llaves serán aquellas añadidas, o cuya frecuencia se aumente, por la modificación introducida por el algoritmo *LeZi Update*, y por lo tanto en su creación se marcan con un asterisco en el árbol correspondiente, denominado árbol LZU, que se muestra en la Figura 2.

En cuanto al cálculo de probabilidades, se propone otra forma de usar el árbol LZU obtenido para hacer predicciones. Esta técnica, basada en el algoritmo PPM (*Prediction by Partial Matching*), funciona como se explica a continuación. En primer lugar hay que fijarse en el último contexto de predicción (subcadena con los últimos símbolos del historial L) más largo, que en este caso es $l = ab$, puesto que el nodo que representa la cadena cab (contexto de predicción de orden inmediatamente superior) no existe en el árbol LZU. Teniendo

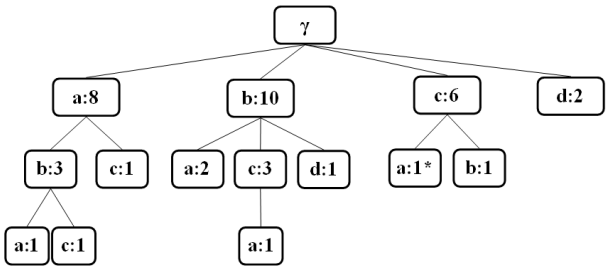


Fig. 2. Árbol LZU tras analizar la cadena del ejemplo

en cuenta este contexto de predicción se puede construir una tabla con la frecuencia de cada símbolo que ha seguido a los contextos de predicción de orden 2 (ab), orden 1 (b) y orden 0 (γ). Para construir dicha tabla hay que tener en cuenta lo que se denominan **eventos de escape**, que se calculan como el número de apariciones de un nodo en las que no está seguido de ningún símbolo. Por ejemplo, la cadena ab tiene una frecuencia igual a 3, pero tiene dos nodos hijo entre los que suman 2 eventos, y por tanto el nodo ab cuenta con un evento de escape. Por otro lado, para contar las apariciones de cada nodo se hace uso de lo que se conoce como **técnica de exclusión**, según la cual hay que tener en cuenta la frecuencia del nodo y la suma de frecuencias de sus nodos hijo. De esta forma, si el nodo ab tiene una frecuencia igual a 3, y la suma de las frecuencias de sus hijos es igual a 2, entonces se considerará que la frecuencia de aparición real de b tras el contexto a es sólo 1. La Tabla I muestra un resumen de las apariciones de los nodos que siguen a los contextos de interés.

Orden	Contexto	Siguiente símbolo				Eventos escape	Eventos totales
		a	c	d	b		
2	ab	1	1			1	3
1	b	2	2	1		4	10
0	γ	4	4	2	4	0	23

Tabla I
FRECUENCIA DE LOS SÍMBOLOS QUE SIGUEN AL CONTEXTO ACTUAL EN EL ALGORITMO *LeZi Update*

Con esta tabla se pueden hallar fácilmente las probabilidades de que cada símbolo conocido sea el correspondiente a la siguiente localización. Por ejemplo, para calcular la probabilidad de que el siguiente símbolo sea c habría que sumar la probabilidad de que el siguiente símbolo del contexto de orden 2 sea una c ($1/3$, 1 evento de 3 totales) más la probabilidad resultante de "escapar" al contexto de orden 1, b, que ocurre con probabilidad $P(es|ab) = 1/3$ (1 evento de escape de 3 totales). La probabilidad de que c siga al contexto de orden 1 es igual a $2/10$ (2 eventos de 10 totales) más la probabilidad que se deriva al considerar el contexto de orden 0, para lo que hay una probabilidad de escape $P(es|b) = 4/10$. En la expresión 2 se puede ver este cálculo en forma matemática y en la expresión 3 se particulariza con los datos del ejemplo.

$$P(c) = P(c|ab) + P(es|ab) \cdot \{P(c|b) + P(es|b) \cdot [P(c|\gamma)]\} \quad (2)$$

$$P(c) = \frac{1}{3} + \frac{1}{3} \cdot \left\{ \frac{2}{10} + \frac{4}{10} \cdot \left[\frac{4}{23} \right] \right\} = 0.423 \quad (3)$$

A pesar de que sigue sin resolverse el problema de las cadenas entre patrones, el algoritmo *LeZi Update* introduce las siguientes ventajas.

- Resuelve el problema de probabilidades cero. En el algoritmo LZ, si un símbolo nunca había seguido al contexto actual, se le asignaba probabilidad cero, mientras que con la técnica de cálculo de probabilidades basada en PPM esta situación se ve resuelta.
- Solventa el problema de las subcadenas dentro de patrones.

C. Algoritmo Active LeZi

Este algoritmo [8] trata de solventar el problema de las cadenas entre patrones. La solución que propone es mantener una ventana de longitud variable de símbolos vistos hasta el momento. La elección de la longitud de la ventana en cada iteración será igual a la de la subcadena s_i más larga vista hasta el momento en el algoritmo LZ clásico.

El funcionamiento es el mismo que el del algoritmo LZ, salvo que tras hacer el análisis que haría dicho algoritmo, se actualiza la ventana con el símbolo actual, y se añaden o incrementan las frecuencias de todas las subcadenas de la ventana que incluyan el símbolo actual. En la Tabla II se muestra la evolución de la ventana y los nodos a añadir o actualizar en las primeras iteraciones del ejemplo inicial. En negrita se indican los nodos que serían creados/actualizados por el algoritmo LZ, y por lo tanto, estos nodos serán los que no estén marcados. Siguiendo la evolución de esta tabla se llega al árbol de la Figura 3, denominado árbol ALZ.

Símbolo	Procesado LZ	Ventana	Añadir/Actualizar
a	Añade a	a	a (**a**)
b	Añade b	b	b (**b**)
a	Nodo a	a	a (a)
b	Añade ab	ab	ab (**a**,**ab**) b (b)
a	Nodo a	ba	ba (b, ba) a (a)
b	Nodo ab	ab	ab (a, ab) b (b)
c	Añade abc	abc	abc (**a**, **ab**, **abc**) bc (b, bc) c (c)
d	Añade d	bcd	bcd (b, bc, bcd) cd (c, cd) d (**d**)
c	Añade c	cdc	cdc (c, cd, cdc) dc (d, dc) c (**c**)

Tabla II
EVOLUCIÓN DE LA VENTANA Y NODOS AFECTADOS EN CADA ITERACIÓN DEL ALGORITMO *Active LeZi* SOBRE EL EJEMPLO INICIAL

Para realizar el cálculo de probabilidades, el algoritmo *Active LeZi* se basa en el algoritmo PPM, aunque en este caso no se aplica la técnica de exclusión. En la Tabla III se muestran las frecuencias de los símbolos que siguen al contexto de predicción ab (el contexto de orden superior, cab, no tiene hijos en el árbol) y las probabilidades se calculan como se indicaba en las expresiones 2 y 3.

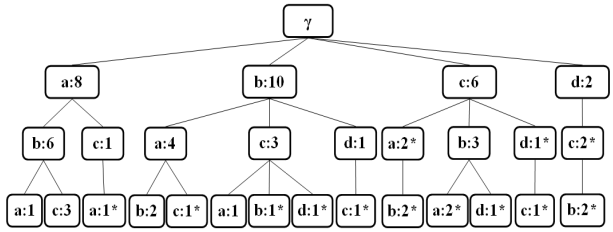


Fig. 3. Árbol ALZ tras analizar la cadena del ejemplo

Orden	Contexto	Siguiente símbolo				Eventos escape	Eventos totales
		a	c	d	b		
2	ab	1	3			2	6
1	b	4	3	1		2	10
0	γ	8	6	2	10	0	26

Tabla III
FRECUENCIA DE LOS SÍMBOLOS QUE SIGUEN AL CONTEXTO ACTUAL

III. IMPLEMENTACIÓN DE LOS ALGORITMOS

La finalidad de los algoritmos de predicción descritos en la Sección II es ejecutarlos en un terminal móvil, de forma que los datos de localización del usuario (LAC y CellID) se vayan procesando en tiempo real según éste se va moviendo. Del procesado de cada nueva celda, los algoritmos obtendrán los símbolos que representan las celdas a las que se desplazará el usuario con mayor probabilidad. Sin embargo, antes de implementar los algoritmos directamente en el terminal, resulta interesante realizar una serie de análisis para comparar probabilidades de acierto, estudiar los efectos de distintas modificaciones de los algoritmos o evaluar el consumo de recursos (memoria y tiempo de ejecución) de cada uno de ellos. De esta forma se pueden evaluar previamente los resultados que se van a obtener y se comprobará si el consumo de recursos permite la ejecución de los algoritmos en dispositivos móviles.

Para realizar estas pruebas se desarrolló una aplicación [9] en lenguaje C cuyo objetivo es realizar el mismo procesado de la información de movilidad que haría un terminal móvil, pero en lugar de procesar cada nueva celda en tiempo real, lo que hace es procesar un fichero de trazas con toda la información de movilidad de cierto período de tiempo recogida previamente por un terminal. Tras el procesado, la aplicación devuelve una serie de ficheros con las predicciones que realizan los algoritmos en cada iteración, es decir, con cada nueva celda que procesan. Comparando estos ficheros de predicciones con el de las trazas reales se puede obtener información sobre la probabilidad de acierto que alcanzan los algoritmos. Además la aplicación también recoge información sobre los recursos consumidos durante el procesado, de forma que también se puedan hacer comparativas sobre este aspecto. La elección del lenguaje C tiene como objetivo hacer el código portable a la aplicación de procesado en teléfonos móviles, desarrollada en Symbian C++ [10]. Esta plataforma dispone de la herramienta OpenC [11], la cual permite integrar código C en una aplicación desarrollada en Symbian C++. Gracias a esta herramienta, el desarrollo del programa para terminales se simplificó enormemente, puesto que sólo era necesario implementar la interfaz gráfica y la captura de datos de la red en tiempo real, integrando directamente todo el código referente a los algoritmos, sin necesidad de reescribirlo.

Es importante hacer hincapié en el tamaño de las trazas. Un día en el que el usuario monitorizado no haya realizado grandes desplazamientos puede dar lugar a una traza de más de 300 símbolos, dado que el método de monitorización no tiene en cuenta si los cambios de celda se hacen por movimiento o sencillamente porque el terminal detecta un nivel mayor de potencia de otra celda, aunque el usuario no se haya movido. Esto supone que el tamaño de los árboles que almacenan el conocimiento adquirido por los algoritmos según procesan nuevos símbolos, así como el tiempo de dicho procesado, llega a alcanzar valores considerables. Por lo tanto el diseño de la aplicación se centró en gestionar de la forma más eficiente posible la memoria ocupada y en tratar de minimizar el tiempo de ejecución de todas las acciones llevadas a cabo durante el procesado de cada símbolo.

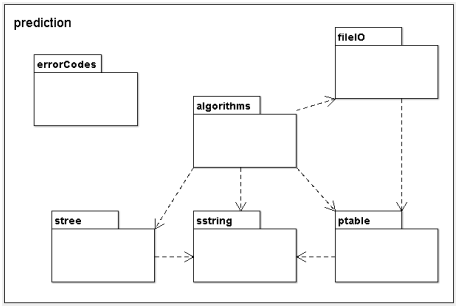


Fig. 4. Arquitectura de la aplicación

La aplicación se organiza tal y como se muestra en la Figura 4. Existen tres módulos básicos que son los encargados de gestionar los contextos y cadenas de símbolos en general (`sstring`), los árboles de símbolos (`stree`) y las tablas de probabilidades (`ptable`). Por su parte, el módulo que engloba todos los algoritmos usados (`algorithms`) usa los tres módulos básicos e interactúa con el módulo de gestión de ficheros (`fileIO`) para leer los símbolos que ha de procesar. Este último bloque es el encargado a su vez de escribir en fichero los resultados derivados del procesado. Todo este funcionamiento está soportado por la aplicación principal (`prediction`) que es la que se encarga de mantener el estado de árboles y contextos durante el análisis de toda la traza, así como de gestionar los posibles errores que se produzcan, los cuales están definidos en su módulo correspondiente (`errorCodes`).

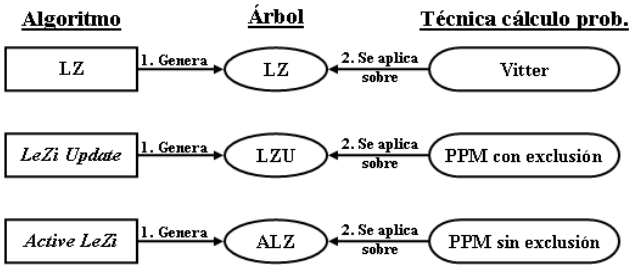
IV. ANÁLISIS DE RESULTADOS

Las dos trazas que se van a analizar tienen características bastante diferentes. El primer conjunto de datos recoge los cambios de celda a lo largo un día. Durante este período, el usuario enciende su terminal en el área de trabajo y permanece allí hasta finalizar su jornada laboral, momento en el que vuelve a su casa. En total consta de 352 pares de datos (LAC,CellID) que se usarán principalmente para comprobar el funcionamiento de los algoritmos y para fijar una primera idea del consumo de recursos de cada uno. Además será el conjunto sobre el que se apliquen los distintos algoritmos modificados, para decidir cuál es la mejor variante.

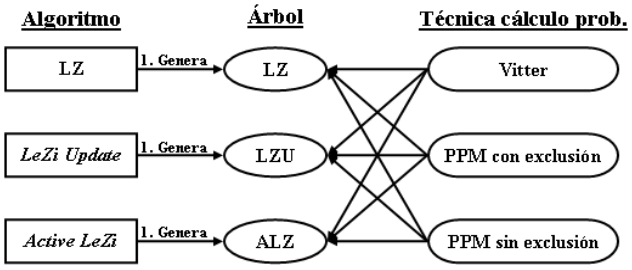
El segundo conjunto de datos recoge la información de dos meses y medio, 31.453 pares de datos en total. En este período de tiempo se incluyen desplazamientos desde la residencia del usuario a su trabajo (medios), desplazamientos dentro del propio área de trabajo (cortos) y viajes entre comunidades dentro de España (largos), con lo que la variedad, tanto de extensión como de velocidad de movimientos, es mucho mayor que en el caso de la primera traza.

A. *Técnicas de cálculo de probabilidades*

Como se comentó en la Sección II, cada uno de los algoritmos explicados contaba con una técnica de cálculo de probabilidades distinta. Sin embargo, resulta interesante analizar cómo se comportan las distintas técnicas para tratar de determinar cuál es la que proporciona mejores resultados, independientemente del algoritmo sobre que el que se aplique. Para ello, se hizo el cálculo con cada una de las técnicas usando los árboles generados por los tres algoritmos, tal y como se muestra en la figura 5.



(a) Teórico, según lo visto en la Sección II



(b) Modificado, para el análisis

Fig. 5. Relación entre los algoritmos y las técnicas de cálculo de probabilidades

En las Figuras 6, 7 y 8 se representa la probabilidad de acierto acumulada en cada iteración, es decir, el número de símbolos acertados entre el número total de símbolos procesados hasta el momento, obtenida por cada una de las variantes. Observando las gráficas se puede comprobar que la probabilidad de acierto alcanzada por la técnica basada en el algoritmo PPM sin exclusión es mejor que la obtenida por cualquiera de las otras dos opciones.

Los resultados conseguidos con los tres algoritmos aplicando la técnica de Vitter son mucho peores, como cabía esperar, ya que el cálculo de probabilidades se realiza con mucha menos información. Hay que tener en cuenta que la característica principal de las técnicas PPM era que mezclaban la información de contextos de varios órdenes y solucionaban el problema de no realizar predicciones cuando es la primera vez que encuentra un contexto de predicción nuevo, tal y como sucedía con la técnica de Vitter.

En cuanto a la comparativa entre las dos técnicas PPM, las diferencias son menores y demuestran que aunque el método de exclusión puede obtener mejores resultados en algún caso, en general es mejor usar la técnica sin exclusión. De momento no se consideran los recursos consumidos ya que no dependen de la técnica de cálculo de probabilidades aplicada, sino que varían con el árbol considerado.

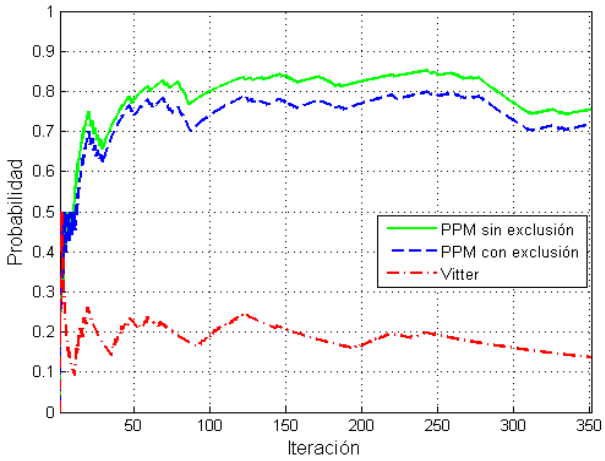


Fig. 6. Evolución de la probabilidad de acierto acumulada para el algoritmo LZ usando distintas técnicas de cálculo de probabilidades

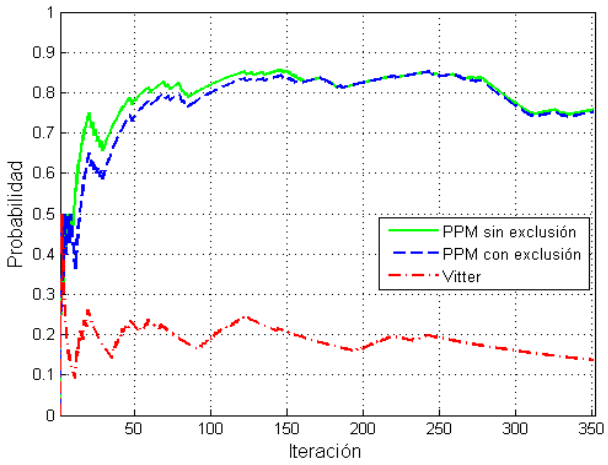


Fig. 7. Evolución de la probabilidad de acierto acumulada para el algoritmo *LeZi Update* usando distintas técnicas de cálculo de probabilidades

B. *Comparativa respecto probabilidad de acierto*

Una vez que se ha visto que la técnica de cálculo de probabilidades que ofrece mejores resultados es aquella basada en el algoritmo PPM sin exclusión, se compararán los resultados obtenidos por los tres algoritmos cuando se les aplica dicha técnica.

En la Figura 9 se muestra la probabilidad de acierto acumulada para cada uno de los algoritmos. Se puede observar que el porcentaje se mantiene en un nivel bastante estable, si se obvian las primeras iteraciones en las que los algoritmos empiezan a aprender y sus resultados son bastante inestables, y exceptuando también breves intervalos en los que la probabilidad desciende ligeramente para volver a subir

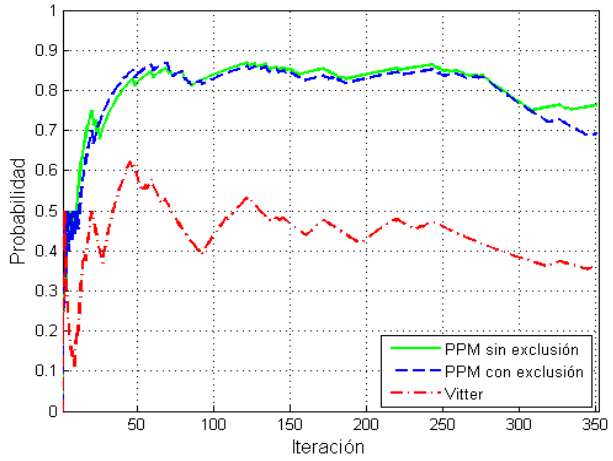


Fig. 8. Evolución de la probabilidad de acierto acumulada para el algoritmo *Active LeZi* usando distintas técnicas de cálculo de probabilidades

pocos símbolos después. Estos fenómenos se relacionan con pequeños desplazamientos dentro de un espacio reducido y a una velocidad no muy elevada, puesto que las pendientes son breves y poco pronunciadas. El evento más significativo tiene lugar hacia el símbolo 270, donde se observa una pendiente que se extiende durante más tiempo. Este hecho indica un desplazamiento a zonas más lejanas de las que se habían visitado hasta el momento, puesto que las cercanas ya se conocen y sin embargo el número de aciertos en este tramo disminuye significativamente. Este evento se corresponde con el intervalo durante el que el usuario regresa a su casa, y por lo tanto transita por zonas por las que no había pasado anteriormente con el dispositivo móvil tomando datos sobre su localización.

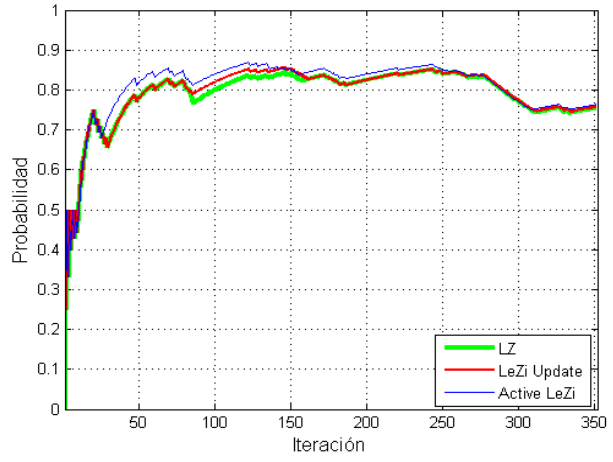


Fig. 9. Probabilidad de acierto acumulada para cada uno de los algoritmos de predicción

No obstante, la Figura 9 no muestra con demasiada objetividad la dimensión de los cambios, puesto que según aumenta el número de iteraciones los cambios van perdiendo relevancia. Esto es debido a que al calcular la probabilidad como el número de aciertos entre el número total de símbolos procesados, el denominador va aumentando con cada nueva iteración, y por lo tanto la importancia de cada nuevo evento va disminuyendo. Para evitar este efecto, se decidió aplicar una ventana a los datos procesados, representando la probabilidad acumulada en los 100 últimos símbolos. Dicha probabilidad, representada en la Figura 10 refleja con mayor claridad los cambios mencionados anteriormente: las pequeñas pendientes hasta la iteración 270 y la gran pendiente a partir de este punto.

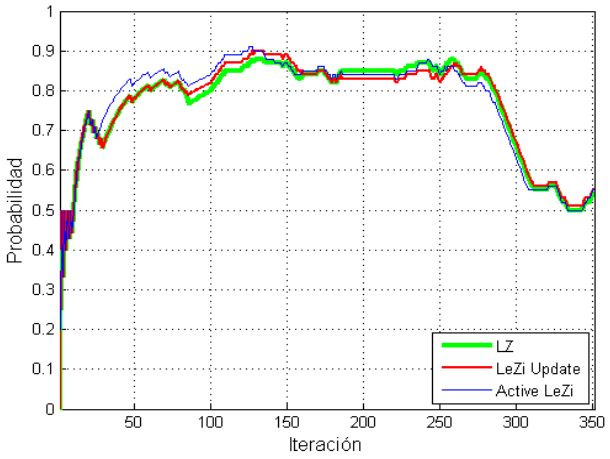


Fig. 10. Probabilidad de acierto parcial para cada uno de los algoritmos de predicción

En cuanto a la capacidad de predicción de cada uno de los algoritmos, se puede observar como el algoritmo *Active LeZi* obtiene mejores resultados en algunos intervalos, pero seguido muy de cerca por los otros dos algoritmos, entre los que apenas hay diferencia.

C. Comparativa respecto recursos consumidos

Una vez analizada la probabilidad de acierto, es interesante fijarse en los recursos consumidos por cada algoritmo. La Tabla IV muestra el tamaño del árbol asociado a cada algoritmo, tanto en número de nodos como en tamaño ocupado en memoria. Se puede comprobar que existe una diferencia apreciable entre los tamaños de los árboles LZ y LZU y el tamaño del árbol ALZ, que tras un día de procesado supera en un orden de magnitud a los otros dos. Esta diferencia se hace incluso más patente al observar los resultados de procesar la traza de dos meses y medio, tras los que el árbol ALZ ocupa más de 100 MB de memoria, valor que puede resultar prohibitivo a la hora de ejecutar este algoritmo en un terminal móvil (y por tanto de memoria bastante limitada) durante un intervalo de tiempo relativamente largo.

Algoritmo	**Tamaño**			
	1 día		**2 meses y medio**	
	Nodos	**Memoria**	**Nodos**	**Memoria**
LZ	99	12 KB	6.000	703 KB
LeZi Update	131	16 KB	8.000	938 KB
Active LeZi	1.326	156 KB	900.000	103 MB

Tabla IV
TAMAÑO DEL ÁRBOL ASOCIADO A CADA UNO DE LOS ALGORITMOS TRAS UN DÍA DE PROCESADO

En cuanto a la memoria consumida por los contextos de predicción, se comprobó que el crecimiento es exactamente el mismo para los tres algoritmos, tal y como cabía esperar.

Hay que recordar que la base de estos algoritmos es que en cada iteración alcanzan el orden óptimo, k, que es el que determina el tamaño del contexto, y por lo tanto dicho tamaño será el mismo en los tres casos. De esta forma se concluye que el tamaño ocupado por este contexto no aporta demasiada información para comparar los algoritmos, puesto que la memoria ocupada en los tres casos será la misma.

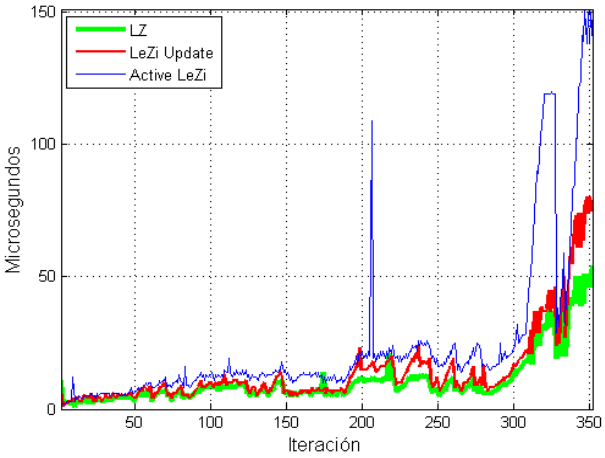


Fig. 11. Evolución del tiempo de procesado de símbolo para cada uno de los algoritmos

Finalmente, el último recurso a tener en cuenta es el tiempo de procesado de cada nuevo símbolo. En este punto es importante aclarar que todas las medidas se han realizado en un ordenador con una velocidad de procesador de 2.2 GHz. A pesar de que la velocidad de procesamiento de un terminal móvil es más limitada, lo que se pretende con estas medidas de tiempo es realizar una comparativa entre algoritmos, y por tanto el valor exacto no es tan importante. La medición de tiempos en el teléfono móvil se realizará en estudios posteriores. En la Figura 11 se representa la evolución de este parámetro. Tal y como cabía esperar de la evolución del tamaño del árbol, también existe una diferencia notable entre los algoritmos LZ y *LeZi Update* y el algoritmo *Active LeZi* . Además esta diferencia se observa sobre todo a partir del símbolo 270, momento en el que el tamaño del árbol también comienza a crecer de forma más acusada, haciendo las búsquedas en el mismo más tediosas. No obstante, en este caso la diferencia entre algoritmos no es tan marcada como en el caso del tamaño de árbol, por lo que el tiempo no resulta un factor tan crítico a la hora de evaluar los algoritmos. Cabe notar que los picos que se distinguen en la gráfica son debidos a que esta aplicación se ejecuta junto con el resto de tareas del sistema operativo, y en algunas ocasiones éstas toman el procesador durante el análisis de un nuevo símbolo, aumentando el tiempo de procesado. De hecho se comprobó que varias ejecuciones del procesado de una misma traza daban como resultado gráficas con diferente número de valores puntuales de pico en distintas iteraciones.

D. Probabilidad de acierto con más de un símbolo

Hasta el momento se ha analizado la precisión alcanzada por los distintos algoritmos si se toma como predicción únicamente el símbolo con mayor probabilidad de ser el correspondiente a la siguiente localización. Sin embargo, en esquemas de reserva de recursos también puede resultar útil tener en cuenta no sólo una celda sino varias para aumentar el porcentaje de aciertos. Por lo tanto se realizaron análisis sobre la probabilidad de acierto obtenida por cada algoritmo si se tomaban como predicción los uno, dos o tres símbolos más probables de ser los correspondientes a la siguiente localización. Los resultados obtenidos se muestran en las Figuras 12, 13 y 14.

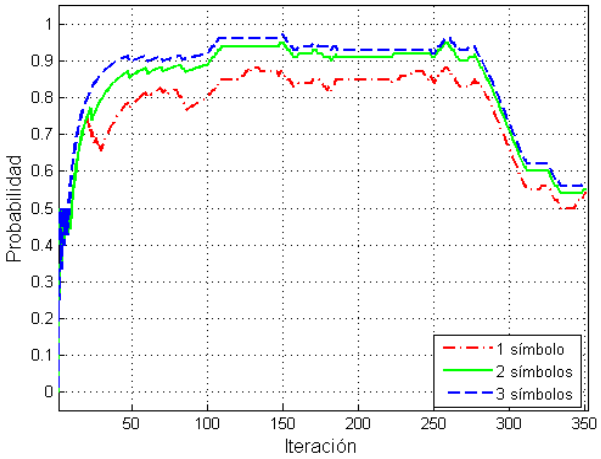


Fig. 12. Evolución de la probabilidad de acierto del algoritmo LZ teniendo en cuenta distintas cantidades de símbolos

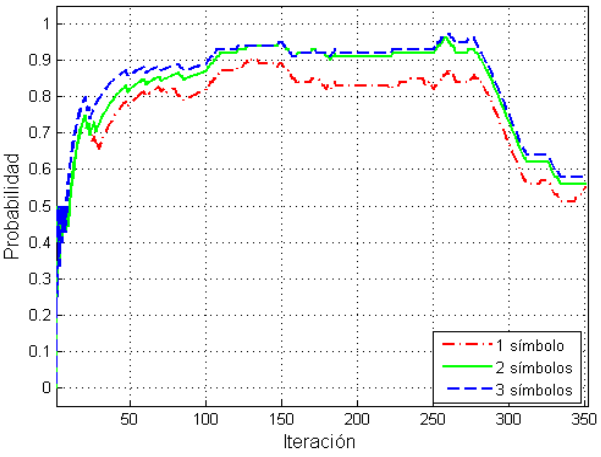


Fig. 13. Evolución de la probabilidad de acierto del algoritmo *LeZi Update* teniendo en cuenta distintas cantidades de símbolos

Como se puede observar, la probabilidad de acierto aumenta notablemente si se usan dos símbolos. Esta mejoría es mayor aún usando tres símbolos, aunque el incremento de la probabilidad de acierto de dos a tres símbolos no es tan pronunciado como el aumento obtenido al pasar de uno a dos símbolos. Cabe recordar que la técnica de cálculo de probabilidades correspondiente (en este caso aquella basada en el algoritmo PPM sin exclusión) devuelve como resultado la lista de posibles siguientes símbolos, ordenada de mayor a menor probabilidad de que cada símbolo sea el siguiente evento del historial de movimientos. Por lo tanto, los recursos consumidos son los mismos independientemente del número de símbolos considerados, razón por la cuál no se muestran gráficas correspondientes a la evolución de los recursos consumidos.

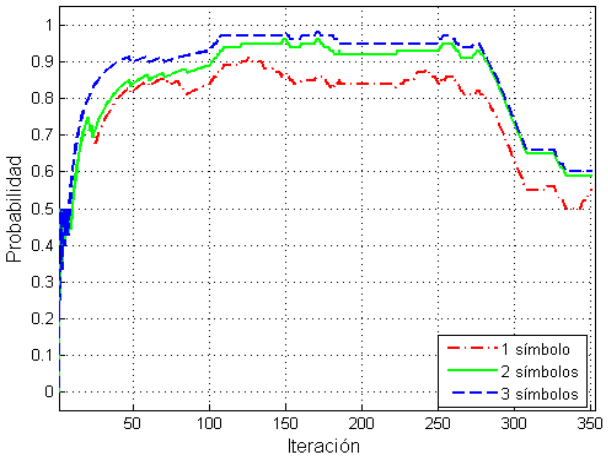


Fig. 14. Evolución de la probabilidad de acierto del algoritmo *Active LeZi* teniendo en cuenta distintas cantidades de símbolos

V. Conclusiones

Para finalizar, se resumen a continuación las conclusiones que se pueden extraer del análisis de prestaciones realizado.

- Ninguno de los tres algoritmos resalta especialmente sobre los demás por su probabilidad de acierto, sino que todos obtienen resultados muy similares.
- Las pequeñas diferencias en la probabilidad de acierto se traducen en tamaños de árbol bien distintos.
- De las tres técnicas de cálculo de probabilidades consideradas, se demostró que la que mejores resultados obtiene es aquella basada en el algoritmo PPM sin exclusión.
- Se comprobó que con el uso de dos símbolos se conseguía una importante mejora de la probabilidad de acierto.

Existen numerosas líneas futuras de investigación posibles que se pueden llevar a cabo a partir de este estudio. Entre ellas cabe destacar las siguientes:

- Estudiar nuevas mejoras de los algoritmos y realizar más análisis de prestaciones con distintas trazas.
- Analizar las mejoras que introduciría el uso de estas técnicas de predicción en los esquemas de reserva de recursos.
- Estudiar técnicas de poda (*prunning*) que traten de minimizar el problema del consumo de recursos.
- Introducir información temporal que permita predecir también cuándo se llegará a la siguiente localización.
- Analizar el consumo de batería que supone la ejecución de estos algoritmos en terminales móviles.

Agradecimientos

Este trabajo ha sido parcialmente soportado por el proyecto España Virtual. España Virtual es un proyecto de I+D, subvencionado por el CDTI dentro del programa Ingenio 2010, orientado a la definición de la arquitectura, protocolos y estándares del futuro Internet 3D, con un foco especial en lo relativo a visualización 3D, inmersión en mundos virtuales, interacción entre usuarios y a la introducción de aspectos semánticos, sin dejar de lado el estudio y maduración de las tecnologías para el procesamiento masivo y almacenamiento de datos geográficos.

Con una duración de cuatro años, el proyecto está liderado por DEIMOS Space y cuenta con la participación del Centro Nacional de Información Geográfica (IGN/CNIG), Grid Systems, Indra Espacio, GeoVirtual, Androme Ibérica, GeoSpatiumLab, DNX y una decena de prestigiosos centros de investigación y universidades nacionales.

Asimismo, este trabajo ha sido parcialmente soportado por el proyecto Integración Vertical de Servicios Telemáticos de Apoyo al Aprendizaje en Entornos Residenciales Distribuidos, CCG08-UC3M/TIC-4479

Referencias

[1] F. Telefónica, *La Sociedad de la Información en España 2008*. Editorial Ariel S.A., 2008.

[2] D. A. Levine, I. F. Akyildiz, and M. Naghshineh, "The shadow cluster concept for resource allocation and call admission in ATM-based wireless networks," in *Proceedings of the 1st Annual International Conference on Mobile Computing and Networking*, (Berkeley, California, United States), pp. 142–150, ACM, Nov 13-15 1995.

[3] A. Bhattacharya and S. K. Das, "LeZi-Update: an information-theoretic approach to track mobile users in PCS networks," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, (Seattle, Washington, United States), ACM, 2002 Kluwer Academic Publishers, August 15-19 1999.

[4] C. Cheng *et al.*, *Wireless Internet Handbook: Technologies, Standards, and Applications*, ch. Location prediction algorithms for mobile wireless systems, pp. 245–263. Boca Raton, FL: B. Furht and M. Ilyas, Eds. CRC Press, 2003.

[5] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp. 530–536, 1978.

[6] L. Song, D. Kotz, R. Jain, and X. He, "Evaluating next-cell predictors with extensive wi-fi mobility data," *IEEE Transactions on Mobile Computing*, vol. 5, pp. 1633–1649, Dec 2006.

[7] J. S. Vitter and P. Krishnan, "Optimal prefetching via data compression," *J. ACM*, vol. 43, pp. 771–793, Sep 1996.

[8] K. Gopalratnam and D. J. Cook, "Online sequential prediction via incremental parsing: the Active LeZi algorithm," *Intelligent Systems, IEEE*, vol. 22, pp. 52–58, Jan.-Feb. 2007.

[9] A. Rodríguez-Carrión, "Estudio e implementación de algoritmos para la predicción de la localización." Proyecto Final de Carrera, Mar 2009.

[10] R. Barker, L. Edwards, and S. of EMCC Software Ltd., *Developing Series 60 Applications: A Guide for Symbian OS C++ Developers*. Addison Wesley, 2004.

[11] F. H. P. Fitzek and F. Reichert, *Mobile Phone Programming and its Application to Wireless Networking*, ch. Open C, pp. 139–158. Springer, 2007.

[12] B. Sun, Z. Chen, R. Wang, Y. Fei, and V. V.C.M., "Towards adaptive anomaly detection in cellular mobile networks," in *Consumer Communications and Networking Conference, 2006. CCNC 2006. 2006 3rd IEEE*, vol. 2, pp. 666–670, IEEE, Jan 8-10 2006.

Diseño intercapas en redes inalámbricas basadas en AMC y ARQ truncado

Jaume Ramis, Guillem Femenias y Loren Carrasco
Grupo de Comunicaciones Móviles - Departamento de Matemáticas e Informática,
Universitat de les Illes Balears
Ctra. Valldemossa, km. 7,5. 07122-Palma.
{jaume.ramis,guillem.femenias,loren.carrasco}@uib.es

***Resumen*—En este artículo se presenta un modelo basado en cadenas de Markov que permite el desarrollo de un diseño intercapas en redes inalámbricas en las que se combina la utilización de un esquema de modulación y codificación adaptativas (AMC, *adaptive modulation and coding*) en la capa física con un protocolo ARQ (*automatic repeat request*) truncado en la capa de enlace de datos. Además, se proponen dos posibles opciones para el desarrollo del esquema AMC, según se considere el valor medio o bien el valor instantáneo de la tasa de error de paquete que se pretenda no sobrepasar a nivel de la capa física. A partir del modelo analítico obtenido se obtienen expresiones cerradas del throughput, retardo medio de paquete y tasa media de pérdida de paquetes debido tanto al desbordamiento del buffer como a haber excedido el número máximo permitido de retransmisiones. Con ello se formula un diseño intercapas planteado como un problema de optimización con restricciones que permite maximizar el throughput garantizando los requisitos de calidad de servicio.**

***Palabras Clave*—diseño intercapas, modulación y codificación adaptativas, ARQ truncado, cadena de Markov, calidad de servicio (QoS)**

I. Introducción

Los diseños intercapas en redes inalámbricas, en los que se permite la interacción e intercambio de información entre distintas capas de la pila de protocolos, han adquirido una creciente popularidad en los últimos años [1]–[3]. Concretamente, un gran número de propuestas coinciden en combinar esquemas de modulación y codificación adaptativas (AMC, *adaptive modulation and coding*) en la capa física con protocolos ARQ (*automatic repeat request*) en la capa de enlace de datos con el objetivo de mejorar la eficiencia espectral haciendo uso de manera conjunta de la adaptabilidad de AMC a las condiciones del canal inalámbrico y de la capacidad correctora de errores de los protocolos ARQ [4]–[12].

Entre los diseños intercapas AMC/ARQ cabe destacar un esquema particularmente interesante, propuesto por Liu *et al.* [4], que combina AMC con un protocolo ARQ truncado con el objetivo de mejorar el throughput medio y simultáneamente reducir el jitter de retardo y/o limitar el tamaño de los buffers del transmisor y del receptor. En este marco, Wang *et al.* [11] y Le *et al.* [12] desarrollaron modelos similares basados en cadenas de Markov con el propósito de analizar el proceso de encolamiento subyacente inducido por la combinación del esquema AMC y del protocolo ARQ truncado. Sin embargo, con el objetivo de facilitar el análisis de colas, ambas propuestas utilizan hipótesis poco realistas. Suponen, en primer lugar, que cada trama de la capa física contiene como máximo un paquete de la capa de enlace de datos y, en segundo lugar, que el transmisor dispone de información instantánea sobre los reconocimientos de recepción (ACK/NACK) en cada trama. Además, estas propuestas se basan en el modelado de la amplitud de los desvanecimientos del canal inalámbrico mediante cadenas de Markov de primer orden con estados finitos. Sin embargo, tal como demostraron Tan y Beaulieu en [13], estos modelos se caracterizan por tener una función de autocorrelación que decrece exponencialmente, incapaz de ajustarse a la función de autocorrelación hipergeométrica propia de los procesos Rayleigh utilizados para modelar los canales inalámbricos con desvanecimientos no selectivos en frecuencia [14], comprometiendo de esta forma el diseño de protocolos de capas superiores.

En este artículo, basándonos en el marco teórico desarrollado en [15], [16], proponemos un nuevo modelo de colas a nivel de enlace que generaliza las herramientas analíticas propuestas por Wang *et al.* [11] y Le *et al.* [12]. Nuestra propuesta, por una parte, permite que cada trama de la capa física contenga el número de paquetes de la capa de enlace de datos que el modo de transmisión de la capa física seleccionado permita transmitir y, por otra, asume que el transmisor dispone de la confirmación de la recepción de los paquetes al final de cada slot TDMA. Además se plantean dos opciones a la hora de desarrollar el esquema AMC, según se considere el valor medio (optimización ergódica) o bien el valor instantáneo (optimización instantánea) de la tasa de error de paquete como criterio de calidad de servicio. Además, a partir de ello, y haciendo uso de una cadena de Markov bidimensional, se derivan expresiones analíticas del throughput, retardo medio de paquete, y tasa media de pérdida de paquetes debido tanto al desbordamiento del buffer como al hecho de haber superado el número máximo permitido de retransmisiones. Este modelo analítico se utiliza para formular un diseño intercapas concebido como un problema de optimización con restricciones para explotar conjuntamente el impacto sobre la calidad de servicio de AMC en la capa física y ARQ en la capa de enlace. Este modelo permite contrastar los resultados obtenidos utilizando ambos métodos de optimización AMC, la ergódica y la instantánea.

El artículo está organizado de la siguiente manera. En la Sección II se define el modelo del sistema y se especifican las hipótesis realizadas. En la Subsección II-A se describe el proceso de generación de paquetes, en la Subsección II-B se especifican los dos esquemas AMC propuestos y en la Subsección II-C se presenta el modelo de Markov bidimensional de primer orden de la capa física. En la Sección III se describe el proceso de encolamiento inducido conjuntamente por el protocolo ARQ truncado y el esquema AMC mediante una

cadena de Markov encastada. Nuestra propuesta de diseño intercapas se presenta en la sección IV y en la sección V se muestran los resultados numéricos obtenidos con la aplicación del modelo analítico propuesto. Para finalizar, las conclusiones, resultados y contribuciones más significativas se sumarizan en la Sección VI.

II. Modelo del sistema e hipótesis

Consideramos un sistema de transmisión de paquetes punto a punto que utiliza AMC en la capa física y un protocolo ARQ truncado en la capa de enlace de datos. La unidad básica de procesamiento en la capa de enlace de datos es un paquete y la unidad de procesamiento en la capa física es una trama. Se supone un enlace que soporta tráfico con QoS garantizada caracterizada por un máximo retardo medio de paquete $D_{p\max}$ y una tasa de pérdida de paquetes objetivo en la capa de enlace $P_{l\max}$. Se dispone de un buffer FIFO (*first-in-first-out*) en el transmisor con capacidad para almacenar hasta un máximo de $\overline{Q}$ paquetes. El máximo número permitido de retransmisiones ARQ es N_r. Los paquetes serán eliminados del buffer en caso de haberse recibido correctamente en el receptor móvil o bien después de haberse superado N_r intentos fallidos de transmisión.

El esquema AMC dispone de un conjunto $\boldsymbol{\mathcal{M}}_p = \{0, \ldots, M_p - 1\}$ de M_p modos de transmisión *posibles*, cada uno de los cuales corresponde a una determinada combinación de modulación y codificación, incluyendo el caso en que el transmisor no transmite. Cuando el sistema utiliza el modo de transmisión $n \in \boldsymbol{\mathcal{M}}_p$, el sistema transmite $p_n = bR_n$ paquetes por trama, donde R_n representa el número de bits de información por símbolo correspondiente al modo de transmisión (TM, *transmission mode*) n y b es un parámetro de diseño que determina el número de paquetes transmitidos por trama. Por conveniencia, se considera $p_0 < \cdots < p_{M_p-1}$, con $p_0 = 0$ (es decir, el modo de transmisión 0 corresponde a la ausencia de transmisión) y $p_{M_p-1} \triangleq \mathcal{C}_p$. Tal y como se muestra en [15], dependiendo de las condiciones del canal y de los requisitos de QoS, algunos de estos M_p modos de transmisión *posibles* pueden ser declarados como *inútiles* y entonces sólo un conjunto $\boldsymbol{\mathcal{M}} = \{0, \ldots, M-1\}$ de M modos *útiles* estará disponible para el esquema AMC. Cuando el sistema utiliza el modo de transmisión *útil* $n \in \boldsymbol{\mathcal{M}}$, el sistema transmite c_n paquetes y, por conveniencia, se considera también que $c_0 < \cdots < c_{M-1}$, con $c_0 \geq 0$ y $c_{M-1} = \mathcal{C} \leq \mathcal{C}_p$. Adoptamos un modelo de canal Rayleigh quasi-estático para describir los desvanecimientos rápidos [17], según el cual el canal se mantiene invariante durante el período de tiempo correspondiente a una trama de T_f segundos[1] y puede variar entre sucesivos intervalos de trama. Se asume que el receptor dispone de información ideal sobre el estado del canal y que, en consecuencia, el controlador AMC del receptor lleva a cabo un proceso ideal de selección del modo de transmisión trama a trama. Además, suponiendo que los mensajes de confirmación de la transmisión son de reducido tamaño y están altamente protegidos mediante codificación FEC (*forward error correction*), se considera un canal de realimentación ARQ instantáneo y libre de errores.

[1]En este artículo se asume que la duración de la trama es menor que el tiempo de coherencia del canal.

II-A. Proceso de generación de paquetes

Como en [8] y [18] se modelará el proceso de generación de paquetes a través de un proceso Markoviano discreto de llegadas por lotes (D-BMAP, *discrete batch Markovian arrival process*). De acuerdo con lo establecido por Blondia en [19], un D-BMAP puede describirse mediante matrices subestocásticas $\boldsymbol{U}_a$, $a = 0, 1, 2, \ldots$, de orden $\mathcal{A} \times \mathcal{A}$, cuyos elementos $u_a(i,j)$ corresponden a la probabilidad de una transición de la fase i a la fase j con un lote de llegadas de tamaño a y $\sum_{a=0}^{\infty}\sum_{j=1}^{\mathcal{A}} u_a(i,j) = 1$. La matriz de probabilidades de transición puede obtenerse como

$$\boldsymbol{U} = \sum_{a=0}^{\infty} \boldsymbol{U}_a. \tag{1}$$

Debido a la propiedad markoviana del proceso de llegadas tenemos que $\boldsymbol{\omega} = \boldsymbol{\omega}\boldsymbol{U}$ y $\boldsymbol{\omega}\mathbf{1}_{\mathcal{A}} = 1$, donde $\boldsymbol{\omega}$ representa el vector de probabilidades estacionarias del D-BMAP y $\mathbf{1}_{\mathcal{A}}$ es un vector columna de longitud $\mathcal{A}$ con todos sus elementos iguales a uno. En este caso la tasa media de llegada λ puede calcularse como

$$\lambda = \boldsymbol{\omega} \sum_{a=0}^{\mathcal{A}-1} a\, \boldsymbol{U}_a \mathbf{1}_{\mathcal{A}}. \tag{2}$$

Un D-BMAP constituye una generalización de un amplio conjunto de procesos usados en el modelado de teletráfico. Por ejemplo, ajustando adecuadamente los parámetros del D-BMAP, puede obtenerse un proceso de Poisson, un proceso de Poisson por lotes, un proceso MMPP (*Markov modulated Poisson process*) o un proceso de llegadas Markoviano. En este artículo, igual que en [8] y [18], se asume que la fuente es un caso especial de proceso de llegadas D-BMAP, que está caracterizado por una matriz $\boldsymbol{U}$ de probabilidades de transición de $\mathcal{A}$ estados, donde en el estado $a \in \{0, \ldots, \mathcal{A}-1\}$, en una unidad temporal se generan a paquetes. Entonces, cada matriz subestocástica $\boldsymbol{U}_a$, $a = 0, \ldots, \mathcal{A}-1$, en este D-BMAP particular, puede construirse manteniendo solamente la fila $(a+1)$-ésima de $\boldsymbol{U}$ y fijando el resto de filas a cero. Por definición del D-BMAP es obvio que $\boldsymbol{U}_a = \mathbf{0}$ para cualquier $a \geq \mathcal{A}$. Todos los resultados presentados en este artículo se obtienen utilizando este caso particular de D-BMAP.

II-B. Modulación y codificación adaptativas (AMC)

Sea γ_ν la relación señal a ruido (SNR, *signal to noise ratio*) instantánea en $t = \nu T_f$, donde T_f es el período de trama. Dado el modelo de canal Rayleigh quasi-estático considerado, γ_ν es una variable aleatoria distribuida exponencialmente cuya función de densidad de probabilidad (pdf) es

$$p_{\gamma_\nu}(\gamma) = (1/\overline{\gamma})\exp(-\gamma/\overline{\gamma}), \quad \gamma \geq 0,$$

donde $\overline{\gamma} = E\{\gamma_\nu\}$ es la SNR media en recepción. Los modos de transmisión disponibles en el esquema AMC corresponden a modulaciones M-QAM con codificación convolucional; han sido adoptados del estándar IEEE 802.11a [20] y se encuentran listados en la Tabla I.

En presencia de ruido Gaussiano blanco aditivo (AWGN), la PER (*packet error rate*) de estos modos de transmisión puede aproximarse mediante la expresión

$$\mathrm{PER}_n(\gamma) \approx \begin{cases} 1 & ,0 \leq \gamma < \gamma_{p_n} \\ a_n \exp\left(-g_n\gamma\right) & ,\gamma \geq \gamma_{p_n} \end{cases}$$

Tabla I
MODOS DE TRANSMISIÓN

	Modo 1	Modo 2	Modo 3	Modo 4	Modo 5
Modulación	BPSK	QPSK	QPSK	16QAM	64QAM
R_c	1/2	1/2	3/4	3/4	3/4
R_n	0.50	1.00	1.50	3.00	4.50
a_n	274.723	90.251	67.618	53.399	35.351
g_n	7.993	3.500	1.688	0.376	0.090
γ_{p_n} (dB)	-1.533	1.094	3.972	10.249	15.978

donde n es el índice correspondiente al modo y a_n, g_n y γ_{p_n}, son los parámetros de ajuste correspondientes a una longitud de paquete de 1080 bits y se encuentran listados en la Tabla I [4].

Dada γ_ν, el objetivo de AMC consiste en seleccionar el TM que maximiza la tasa de transmisión de datos asegurando una PER inferior a un valor prescrito P_0. En este artículo planteamos dos posibilidades a la hora de desarrollar el esquema AMC, según se considere el valor medio de PER o bien su valor instantáneo. En ambos casos, y de acuerdo con [6] y [4] respectivamente, se subdivide el rango completo de SNR en un conjunto de intervalos no solapados definido por la partición $\mathbf{\Gamma}^m = \{[\,\gamma_0^m, \gamma_1^m\,), [\,\gamma_1^m, \gamma_2^m\,), \ldots, [\,\gamma_{M-1}^m, \gamma_M^m\,)\}$ con $\gamma_0^m = 0$ y $\gamma_M^m = \infty$, y el modo n será seleccionado cuando $\gamma_\nu \in \left[\gamma_n^m, \gamma_{n+1}^m\right)$.

En el primero de estos dos supuestos, al cual nos referiremos como *opción 1*, en el que se impone que la tasa media de error de paquete $\overline{\text{PER}}_n$ debe ser inferior al valor P_0, la partición $\mathbf{\Gamma}^m$ se obtiene utilizando el algoritmo de búsqueda de umbrales propuesto por Ramis *et al.* [15], [16]. Dicho algoritmo tiene la capacidad de distinguir entre modos de transmisión *útiles* e *inútiles*, garantizando que la PER media satisface la restricción prescrita. Por el contrario, en el segundo caso, al que nos referiremos como *opción 2*, en el cual se requiere que el valor instantáneo de la tasa de error de paquete $\text{PER}_n(\gamma)$ se halle por debajo de P_0, los umbrales correspondientes que definen la partición $\mathbf{\Gamma}^m$ se determinan como la SNR mínima necesaria para garantizar una PER instantánea igual a P_0. De esta manera, invirtiendo la expresión aproximada de $\text{PER}_n(\gamma)$ indicada anteriormente obtenemos

$$\gamma_n^m = \tfrac{1}{g_n} \ln\left(\tfrac{a_n}{P_0}\right) \quad , n = 1, \ldots, M-1 \tag{3}$$

II-C. Modelo de Markov bidimensional del canal

Consideremos la SNR instantánea γ_ν, así como el valor de $\delta_\nu = \gamma_{\nu-1} - \gamma_\nu$. Subdividamos además el rango de γ_ν y δ_ν en un conjunto de celdas bidimensionales no solapadas definidas por las particiones $\mathbf{\Gamma}^c = \left\{[\gamma_0^c, \gamma_1^c), [\gamma_1^c, \gamma_2^c), \ldots, \left[\gamma_{K-1}^c, \gamma_K^c\right)\right\}$ con $\gamma_0^c = 0$ y $\gamma_K^c = \infty$, y $\mathbf{\Delta} = \{(-\infty, 0), [0, \infty)\}$, respectivamente. Entonces podemos modelar el canal mediante un modelo de Markov bidimensional de primer orden, en el cual cada estado corresponde a una de las celdas anteriormente descritas. Es decir, el estado de la cadena de Markov del canal en el instante $t = \nu T_f$ puede expresarse como $\boldsymbol{\zeta}_\nu = (\chi_\nu, \Delta_\nu)$, $\nu = 0, 1, \ldots, \infty$, donde $\chi_\nu = k$ si y sólo si $\gamma_k^c < \gamma_\nu \leq \gamma_{k+1}^c$ y $\Delta_\nu = 0$ (o bien $\Delta_\nu = 1$) si y sólo si $\delta_\nu < 0$ (o $\delta_\nu \geq 0$).

En nuestra propuesta la partición $\mathbf{\Gamma}^c$ se obtiene utilizando el algoritmo de Max-Lloyd [21], [22], desarrollado para el diseño óptimo de cuantificadores no uniformes. De esta manera se determina la partición óptima en cuanto a que se minimiza el error cuadrático medio entre γ_ν y la salida del cuantificador.

II-D. Modelo de Markov bidimensional de la capa física

Tras haber llevado a cabo el diseño del esquema AMC y del modelo de Markov bidimensional de primer orden del canal, subdividamos el rango de γ_ν en un conjunto de intervalos no solapados definidos por la partición $\mathbf{\Gamma}^{m,c} = \left\{[\gamma_0^{m,c}, \gamma_1^{m,c}), [\gamma_1^{m,c}, \gamma_2^{m,c}), \ldots, \left[\gamma_{N_{\text{PHY}}-1}^{m,c}, \gamma_{N_{\text{PHY}}}^{m,c}\right)\right\}$ con $\gamma_0^{m,c} = 0$ y $\gamma_{N_{\text{PHY}}}^{m,c} = \infty$, donde cada intervalo de la partición $\left[\gamma_k^{m,c}, \gamma_{k+1}^{m,c}\right)$ se caracteriza por una determinada combinación de modo de transmisión y estado del canal. De la misma manera que en la Subsección II-C, subdividamos el rango de δ_ν en el conjunto de intervalos no solapados $\mathbf{\Delta} = \{(-\infty, 0), [0, \infty)\}$. Haciendo uso de esta partición bidimensional, podemos definir un modelo de Markov bidimensional de primer orden para la capa física, en el cual cada estado corresponde a una de estas celdas rectangulares bidimensionales. Además, el estado de la cadena de Markov de la capa física en el instante $t = \nu T_f$ puede expresarse como $\boldsymbol{\varsigma}_\nu = (\varphi_\nu, \Delta_\nu)$, $\nu = 0, 1, \ldots, \infty$, donde $\varphi_\nu \in \{0, \ldots, N_{\text{PHY}} - 1\}$ hace referencia a la combinación del modo de transmisión y el estado del canal en dicho período de trama y $\Delta_\nu \in \{0, 1\}$ refleja la característica *up* o *down*[2] de la SNR instantánea en el período de trama $t = (\nu - 1)T_f$.

En cierto instante de tiempo $t = \nu T_f$ el estado de la capa física se caracteriza de manera unívoca por un número entero $n_\nu = 2\varphi_\nu + \Delta_\nu$, con $n_\nu \in \{0, \ldots, 2N_{\text{PHY}} - 1\}$. La capa física se hallará en un estado $n \in \{0, \ldots, 2N_{\text{PHY}} - 1\}$ con una probabilidad estacionaria $P^{\text{PHY}}(n)$ y cada uno de estos estados se caracterizará por una tasa media condicionada de error por paquete $\overline{PER}_n^{\text{PHY}}$. Además, la cadena de Markov de estados finitos de la capa física estará caracterizada por una matriz de probabilidades de transición $\boldsymbol{P}_s = [P_{i,j}]_{0 \leq i,j \leq 2N_{\text{PHY}}-1}$. En este artículo, las probabilidades estacionarias, las tasas medias condicionadas de error por paquete y las probabilidades de transición entre estados, han sido obtenidas o bien numéricamente o bien por simulación utilizando el modelo estadístico de Clarke para los desvanecimientos Rayleigh [14], caracterizado por una frecuencia Doppler normalizada máxima $f_d T_f$.

III. MODELO DE COLAS Y ANÁLISIS

III-A. Cadena de Markov

El proceso de encolamiento inducido conjuntamente por el protocolo ARQ y el esquema AMC puede formularse de manera discreta considerando que una unidad temporal corresponde a un período de trama. Los estados del sistema se observan al inicio de cada unidad de tiempo. Sea $\sigma_\nu = (\mathbf{q}_\nu, a_\nu, \varphi_\nu, \Delta_\nu)$ el estado del sistema en el instante $t = \nu T_f$, donde $\mathbf{q}_\nu = (q_{\nu,0}, \ldots, q_{\nu,N_r})$ representa el estado de la cola en dicho instante, siendo $q_{\nu,i}$ el número de paquetes en la cola que ya han sido transmitidos i veces y $Q_\nu \triangleq \sum_{i=0}^{N_r} q_{\nu,i} \in \{0, \ldots, \overline{Q}\}$ el número total de paquetes

[2]Si $\gamma_\nu < \gamma_{\nu-1}$ la SNR instantánea es decreciente y puede etiquetarse como *down*; por el contrario, si $\gamma_\nu \geq \gamma_{\nu-1}$ entonces la SNR instantánea es creciente y puede etiquetarse como *up*.

en la cola, $a_\nu \in \{0,\ldots,\mathcal{A}-1\}$ representa la fase del D-BMAP, $\varphi_\nu \in \{0,\ldots,N_{\text{PHY}}-1\}$ indica la combinación del modo de transmisión y estado del canal en este intervalo de trama y $\Delta_\nu \in \{0,1\}$ corresponde a la característica *up* o *down* de la SNR instantánea en el período de trama $t=(\nu-1)T_f$. Si nos fijamos en el conjunto de instantes de tiempo $t=\nu T_f$, $\nu=0,1,\ldots,\infty$, las transiciones entre estados son Markovianas. Por ello, se puede utilizar una Cadena de Markov para describir el proceso de encolamiento subyacente. El espacio de estados de esta cadena de Markov finita es $\boldsymbol{\mathcal{S}}=\{\mathcal{S}_n\}_{n=1}^{N_s}$ de dimensión

$$N_s = 2N_{\text{PHY}}\mathcal{A}\sum_{k=0}^{\overline{Q}}\binom{k+N_r}{k}.$$

La probabilidad de transición desde el estado $\mathcal{S}_\mu=(\mathbf{q}_\mu,a_\mu,n_\mu)\in\mathcal{S}$ al estado $\mathcal{S}_{\mu'}=(\mathbf{q}_{\mu'},a_{\mu'},n_{\mu'})\in\mathcal{S}$, donde $n_\mu=2\varphi_\mu+\Delta_\mu$ y $n_{\mu'}=2\varphi_{\mu'}+\Delta_{\mu'}$, puede escribirse como

$$P_{\mathcal{S}_\mu,\mathcal{S}_{\mu'}} = P_{a_\mu,a_{\mu'}}P_{n_\mu,n_{\mu'}}P_{\mathbf{q}_\mu,\mathbf{q}_{\mu'}|a_\mu,n_\mu}$$

donde $P_{a_\mu,a_{\mu'}}$ corresponde a la probabilidad de transición entre las fases del D-BMAP a_μ y $a_{\mu'}$, que puede obtenerse a partir de la matriz $\boldsymbol{U}$, $P_{n_\mu,n_{\mu'}}$ es la probabilidad de transición entre los estados de la capa física n_μ y $n_{\mu'}$, que puede derivarse a partir de la matriz $\boldsymbol{P}_s$ y $P_{\mathbf{q}_\mu,\mathbf{q}_{\mu'}|a_\mu,n_\mu}=\prod_{i=0}^{N_r}P_{q_{\mu,i},q_{\mu',i}|a_\mu,n_\mu}$ corresponde a la probabilidad de transición de la cola desde el estado $\mathbf{q}_\mu$ al estado $\mathbf{q}_{\mu'}$ cuando el D-BMAP se halla en la fase a_μ y la capa física se encuentra en el estado n_μ.

Considérese que el sistema se encuentra en el estado $\mathcal{S}_\mu$ y que $Q_{\mu,i}=\sum_{l=i}^{N_r}q_{\mu,l}$ representa el número de paquetes en la cola que ya han sido transmitidos i o más veces (obviamente, $Q_{\mu,0}=Q_\mu$). Sean, en estas condiciones, $\tau_\mu=\min\left\{Q_\mu,c_{n_\mu}\right\}$ el número de paquetes transmitidos, $\tau_{\mu,i}=\min\left\{q_{\mu,i},\tau_\mu-Q_{\mu,i+1}\right\}$ el número de paquetes transmitidos de entre los que ya habían sido transmidos i veces y $\epsilon_{\mu,i}$ el número de paquetes transmitidos erróneamente de entre los que ya habían sido transmidos i veces. Utilizando esta notación, las únicas transiciones posibles de la cola son

$$q_{\mu',N_r}=\begin{cases} q_{\mu,N_r}-\tau_\mu & ,\tau_\mu\le q_{\mu,N_r}\\ \epsilon_{\mu,N_r-1} & ,\tau_\mu>q_{\mu,N_r}\end{cases}$$

$$q_{\mu',i}=\begin{cases} q_{\mu,i} & ,\tau_\mu\le Q_{\mu,i+1}\\ Q_{\mu,i}-\tau_\mu & ,Q_{\mu,i+1}<\tau_\mu\le Q_{\mu,i}\\ \epsilon_{\mu,i-1} & ,\tau_\mu>Q_{\mu,i}\end{cases}$$

para $i\in\{1,\ldots,N_r-1\}$, y

$$q_{\mu',0}=\begin{cases}\min\left\{\overline{Q}-Q_{\mu',1},q_{\mu,0}+a_\mu\right\} & ,\tau_\mu\le Q_{\mu,1}\\ \min\left\{\overline{Q}-Q_{\mu',1},Q_\mu+a_\mu-\tau_\mu\right\} & ,Q_{\mu,1}<\tau_\mu\le Q_\mu.\end{cases}$$

En consecuencia, si definimos $\mathcal{P}_y^x(z)\triangleq\binom{x}{y}z^y(1-z)^{x-y}$, las probabilidades de transición entre estados pueden obtenerse como

$$P_{q_{\mu,N_r},q_{\mu',N_r}|a_\mu,n_\mu}=\begin{cases}1 & ,\ q_{\mu',N_r}=q_{\mu,N_r}-\tau_\mu\\ & \ \tau_\mu\le q_{\mu,N_r}\\ \mathcal{P}_{q_{\mu',N_r}}^{\tau_{\mu,N_r-1}}\left(\overline{PER}_{n_\mu}^{\text{PHY}}\right) & ,\ \tau_\mu>q_{\mu,N_r}\\ & \ \tau_{\mu,N_r-1}\ge q_{\mu',N_r}\\ 0 & ,\text{ en otro caso}\end{cases}$$

$$P_{q_{\mu,i},q_{\mu',i}|a_\mu,n_\mu}=\begin{cases}1 & ,\ q_{\mu,i}=q_{\mu',i}\\ & \ \tau_\mu\le Q_{\mu,i+1}\\ 1 & ,\ Q_{\mu,i+1}<\tau_\mu\le Q_{\mu,i}\\ & \ q_{\mu',i}=Q_{\mu,i}-\tau_\mu\\ \mathcal{P}_{q_{\mu',i}}^{\tau_{\mu,i-1}}\left(\overline{PER}_{n_\mu}^{\text{PHY}}\right) & ,\ \tau_\mu>Q_{\mu,i}\\ & \ \tau_{\mu,i-1}\ge q_{\mu',i}\\ 0 & ,\text{ en otro caso}\end{cases}$$

para $i\in\{1,\ldots,N_r-1\}$, y

$$P_{q_{\mu,0},q_{\mu',0}|a_\mu,n_\mu}=\begin{cases}1 & ,\ \tau_\mu\le Q_{\mu,1}\\ & \ q_{\mu',0}=\min\left\{\overline{Q}-Q_{\mu',1},q_{\mu,0}+a_\mu\right\}\\ 1 & ,\ Q_{\mu,1}<\tau_\mu\le Q_\mu\\ & \ q_{\mu',0}=\min\left\{\overline{Q}-Q_{\mu',1},Q_\mu+a_\mu-\tau_\mu\right\}\\ 0 & ,\text{ en otro caso.}\end{cases}$$

Con el objeto de obtener los parámetros de análisis del comportamiento del sistema, necesitamos hallar el vector de probabilidades estacionarias. Dicho vector puede calcularse a partir de la matriz de probabilidades de transición $\boldsymbol{P}$ y del vector de probabilidades estacionarias $\boldsymbol{\pi}=\begin{bmatrix}\pi_{\mathcal{S}_1} & \cdots & \pi_{\mathcal{S}_{N_s}}\end{bmatrix}$, dado que cumplen $\boldsymbol{\pi P}=\boldsymbol{\pi}$, imponiendo además la condición de normalización $\boldsymbol{\pi}\mathbf{1}_{N_s}=1$.

III-B. Tasa de pérdida de paquetes y throughput

El número de paquetes perdidos debido al desbordamiento del buffer cuando el estado del sistema cambia desde $\mathcal{S}_\mu$ a $\mathcal{S}_{\mu'}$ puede expresarse como

$$N_{l_{BO}|\mathcal{S}_\mu,\mathcal{S}_{\mu'}}=\begin{cases}\max\{0,Q_{\mu',1}+q_{\mu,0}+a_\mu-\overline{Q}\} & ,\tau_\mu\le Q_{\mu,1}\\ \max\{0,Q_{\mu',1}+Q_\mu-\tau_\mu+a_\mu-\overline{Q}\} & ,\tau_\mu>Q_{\mu,1}.\end{cases}$$

Entonces, el número medio de paquetes perdidos debido al desbordamiento del buffer puede calcularse como

$$\bar{N}_{l_{BO}}=\sum_{\mu=1}^{N_s}\sum_{\mu'=1}^{N_s}\pi_{\mathcal{S}_\mu}P_{\mathcal{S}_\mu,\mathcal{S}_{\mu'}}N_{l_{BO}|\mathcal{S}_\mu,\mathcal{S}_{\mu'}}$$

y la tasa de pérdida de paquetes $P_{l_{BO}}$ puede calcularse como el cociente entre el número medio de paquetes perdidos por desbordamiento del buffer $N_{l_{BO}}$ y el número medio de paquetes entrantes λ en un período de trama, es decir

$$P_{l_{BO}}=\bar{N}_{l_{BO}}/\lambda.$$

El número de paquetes perdidos por el hecho de haber excedido el máximo número permitido de retransmisiones cuando el estado del sistema es $\mathcal{S}_\mu$ puede calcularse como

$$N_{l_{\text{ARQ}}|\mathcal{S}_\mu}=\sum_{l=0}^{\tau_{\mu,N_r}}l\ \mathcal{P}_l^{\tau_{\mu,N_r}}\left(\overline{PER}_{n_\mu}^{\text{PHY}}\right)$$

con $\tau_{\mu,N_r}=\min\{q_{\mu,N_r},c_n\}$, y el correspondiente número medio de paquetes perdidos se obtiene entonces como

$$\bar{N}_{l_{\text{ARQ}}}=\sum_{\mu=1}^{N_s}\pi_{\mathcal{S}_\mu}N_{l_{\text{ARQ}}|\mathcal{S}_\mu}.$$

En consecuencia, la probabilidad de pérdida de paquete debido a haber excedido N_r retransmisiones puede expresarse como

$$P_{l_{\text{ARQ}}}=\bar{N}_{l_{\text{ARQ}}}/\lambda.$$

En este sistema con buffer finito y control de errores basado en ARQ truncado, la tasa de pérdida de paquetes P_l (medida en paquetes por trama) puede expresarse como

$$P_l = P_{l_{\text{BO}}} + P_{l_{\text{ARQ}}}$$

y dada la tasa de pérdida de paquetes P_l, el throughput medio puede calcularse como

$$\eta = \lambda(1 - P_l).$$

III-C. Longitud media de la cola y retardo medio de paquete

Utilizando la fórmula de Little [23], el retardo medio de la cadena de Markov puede calcularse como

$$D_p = L_q/\lambda(1 - P_l) = L_q/\eta,$$

donde L_q representa el número medio de paquetes en la cola, que puede obtenerse como

$$L_q = \sum_{\mu=1}^{N_s} \pi_{\mathcal{S}_\mu} Q_\mu.$$

IV. Diseño intercapas

Dada una longitud máxima disponible de cola $\overline{Q}$, una SNR media $\bar{\gamma}$ y una frecuencia Doppler normalizada máxima $f_d T_f$, las expresiones analíticas obtenidas para el modelo de la capa física dependen básicamente de la PER prescrita P_0, que es un número real en el rango $\mathbf{\Phi} = [0, 1]$ [11]. Entonces, si el sistema debe soportar tráfico con garantías de QoS caracterizado por una tasa de pérdida de paquetes máxima $P_{l\,\text{máx}}$ y un retardo medio de paquete máximo $D_{p\,\text{máx}}$, la propuesta de diseño intercapas debe determinar de manera óptima el valor de la PER prescrita P_0 en la capa física que maximiza el throughput del sistema η, és decir,

$$P_0^{\text{opt}} = \arg \max_{P_0 \in \mathbf{\Phi}} \eta$$

sujeto a las restricciones

$$P_l \leq P_{l\,\text{máx}}$$

$$D_p \leq D_{p\,\text{máx}}.$$

La expresión analítica cerrada de η no deja mucho margen para el desarrollo de algoritmos eficientes para la resolución del problema propuesto de optimización con restricciones. De todos modos, tal como establece Wang *et al.* en [11], dado que los valores de P_0 se hallan en un espacio acotado $\mathbf{\Phi}$, podemos recurrir a una búsqueda exhaustiva para resolver numéricamente el problema de optimización intercapas propuesto.

V. Resultados numéricos

A no ser que se especifique lo contrario, los resultados numéricos serán obtenidos con los siguientes parámetros por defecto: una frecuencia Doppler normalizada máxima $f_d T_f = 0{,}02$, una SNR recibida media $\overline{\gamma} = 8$ dB, un buffer de longitud $\overline{Q} = 5$, un número de estados del canal $K = 10$, un parámetro $b = 2$ y un D-BMAP o bien caracterizado por la matriz de probabilidades de transición [11]

$$\boldsymbol{U} = \begin{bmatrix} 0{,}8 & 0{,}1 & 0{,}05 & 0{,}05 \\ 0{,}05 & 0{,}8 & 0{,}1 & 0{,}05 \\ 0{,}05 & 0{,}05 & 0{,}8 & 0{,}1 \\ 0{,}05 & 0{,}05 & 0{,}1 & 0{,}8 \end{bmatrix} \tag{4}$$

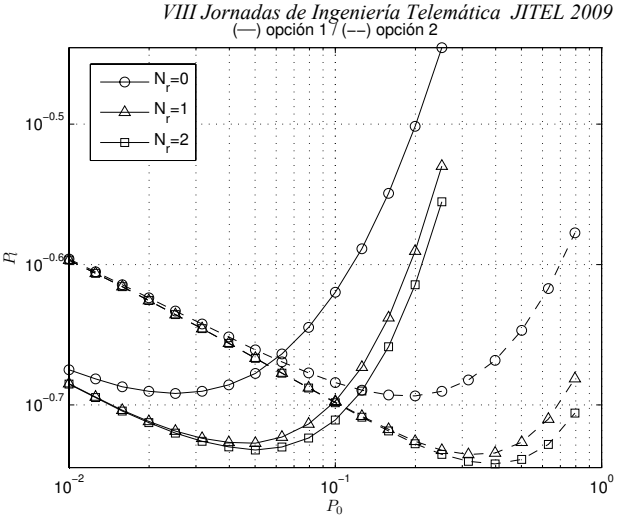


Figura 1. Tasa de pérdida de paquetes vs. P_0.

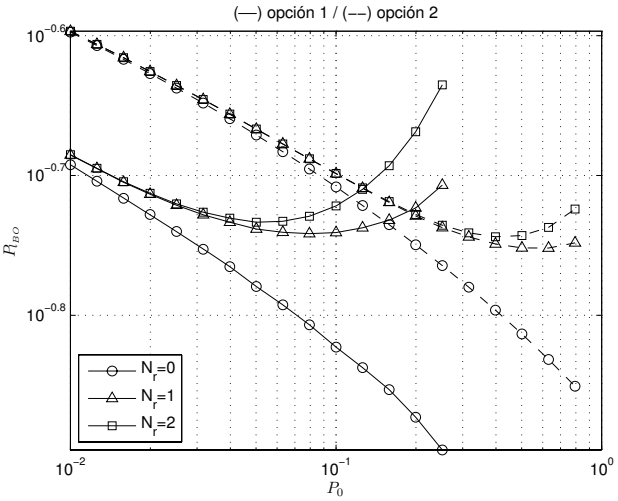


Figura 2. Tasa de pérdida de paquetes por desbordamiento vs. P_0.

o parametrizada para obtener un proceso de Poisson truncado con una tasa de llegadas variable λ necesaria para llevar a cabo el proceso de optimización.

V-A. Dependencia de la tasa de pérdida de paquetes respecto a la PER objetivo

La Figura 1 muestra la dependencia de la tasa media de pérdida de paquetes P_l respecto a la PER objetivo P_0, mientras que las Figuras 2 y 3 detallan el comportamiento de las dos componentes de pérdidas $P_{l_{BO}}$ y $P_{l_{ARQ}}$. Se distinguen las curvas correspondientes a las dos opciones propuestas en la Subsección II-B para la determinación de los umbrales de los modos de transmisión. A partir del análisis de estos gráficos queda claro que, independientemente de que la opción considerada sea la *opción 1* (representada en línea contínua) o la *2* (representada en línea discontinua), para los valores de interés de P_0, la pérdida de paquetes debido al desbordamiento del buffer $P_{l_{BO}}$ es el término de pérdidas dominante. El gráfico de $P_{l_{BO}}$ revela que un incremento de P_0, que conlleva la utilización de modos de transmisión de mayor orden, provoca un incremento de la tasa de servicio y, en consecuencia, una reducción en la probabilidad de desbordamiento del buffer

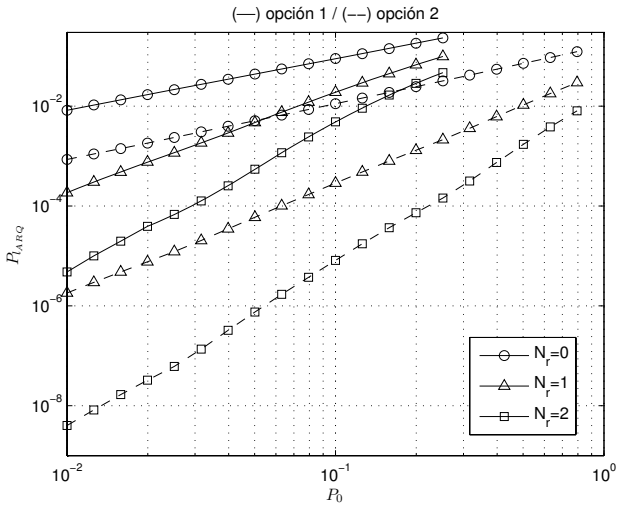


Figura 3. Tasa de pérdida de paquetes por superar N_r vs. P_0.

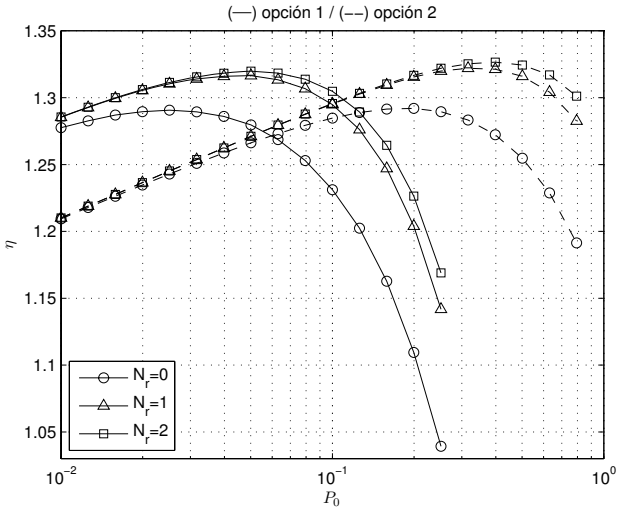


Figura 4. Throughput medio vs. P_0.

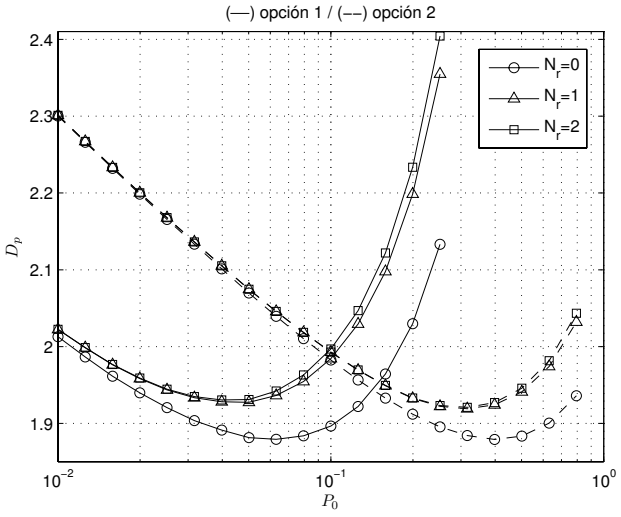


Figura 5. Retardo medio vs. P_0.

$P_{l_{BO}}$. De todas formas, exceptuando el caso en el que las retransmisiones no están permitidas, cuando el aumento de la tasa de servicio no puede hacer frente al gran número de retransmisiones requeridas, $P_{l_{BO}}$ crece rápidamente. En cuanto al comportamiento de las pérdidas por superar el máximo número permitido de retransmisiones $P_{l_{ARQ}}$, el uso de modos de transmisión de mayor orden a medida que aumenta P_0, se traduce en un incremento del número de retransmisiones necesarias para lograr una recepción exitosa y, por consiguiente, en un aumento de $P_{l_{ARQ}}$. Se observa además que valores más altos de N_r implican una disminución de $P_{l_{ARQ}}$ pero, al mismo tiempo, un incremento en el número de paquetes que permanecen en la cola y, en consecuencia, de $P_{l_{BO}}$. Cabe notar que las curvas de pérdidas correspondientes a las dos opciones consideradas para la determinación del esquema AMC reflejan un comportamiento similar, aunque para diferentes valores de P_0. Ello es debido al hecho de que la primera opción, en la que se exige que la tasa media de error de paquete sea inferior al valor prescrito P_0, es menos restrictiva que la segunda, en la cual el nivel de exigencia es superior puesto que se requiere una tasa instantánea de error de paquete inferior al valor P_0. Esto se traduce en la utilización de TMs más robustos y de menor tasa en el segundo esquema, lo que implica que los paquetes perdidos por superar el máximo número permitido de retransmisiones es menor para el esquema AMC correspondiente a esta opción, tal como confirman las curvas de $P_{l_{ARQ}}$. Es por ello que el valor de P_0 en el cual la tendencia decreciente de $P_{l_{BO}}$ se invierte, es menor para la *opción 1*, en que la utilización de TMs de mayor orden provoca que el desbordamiento del buffer debido al gran número de transmisiones fallidas se produzca para valores menores de la PER objetivo.

V-B. Dependencia del throughput y retardo medios respecto a la PER objetivo

Tal y como cabía esperar, las Figuras 4 y 5 muestran, respectivamente, un incremento en el trhoughput η así como en el retardo D_p con el número de retransmisiones permitidas; de todos modos la ganancia más significativa se obtiene al pasar de $N_r = 0$ a $N_r = 1$. El uso de valores mayores de N_r no parece rentable. Estos resultados están en consonancia con los obtenidos en [4]. Asimismo se confirma la analogía en el comportamiento de las dos opciones consideradas para el esquema AMC.

V-C. Ejemplos de diseños intercapas

A título de ejemplo de aplicación del proceso de diseño intercapas presentado en la Sección IV, y con el objetivo de analizar el comportamiento del sistema en función de la tasa media de llegada de paquetes λ, se ha considerado un sistema con una longitud máxima de la cola $\overline{Q} = 10$ paquetes y una SNR media $\overline{\gamma} = 12$ dB. Se han determinado los valores de P_0^{opt} para un tráfico con garantías de QoS caracterizado por una tasa media de pérdida de paquetes máxima $P_{l\,\text{máx}} = 0{,}01$ paquetes/trama y un retardo medio de paquete máximo $D_{p\,\text{máx}} = 2$ tramas. El tráfico ha sido generado utilizando un proceso de Poisson truncado con una longitud de truncamiento igual a 3 paquetes. En la Figura 6 se hallan representados los valores obtenidos para el throughput medio η, la tasa media de pérdida de paquetes P_l, el retardo medio de paquete (en

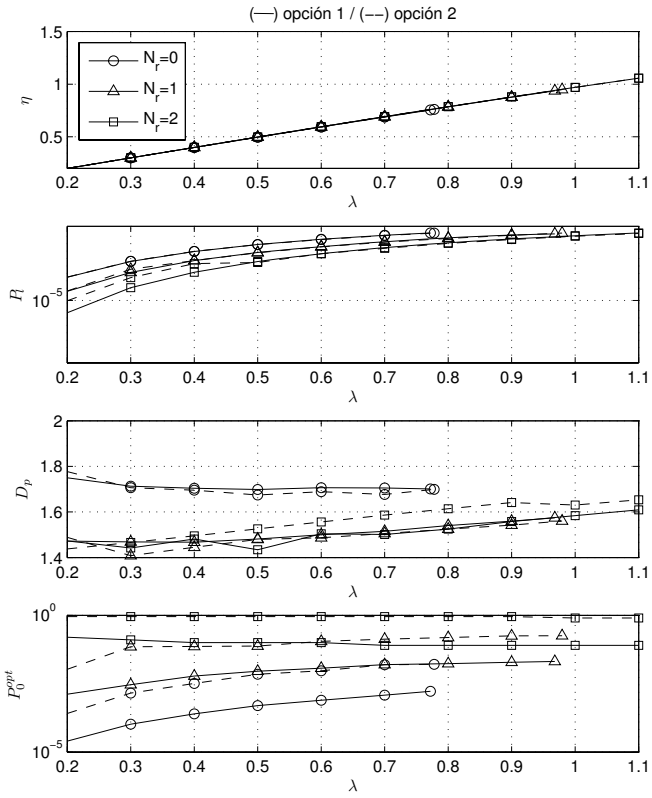


Figura 6. Optimización en función de λ.

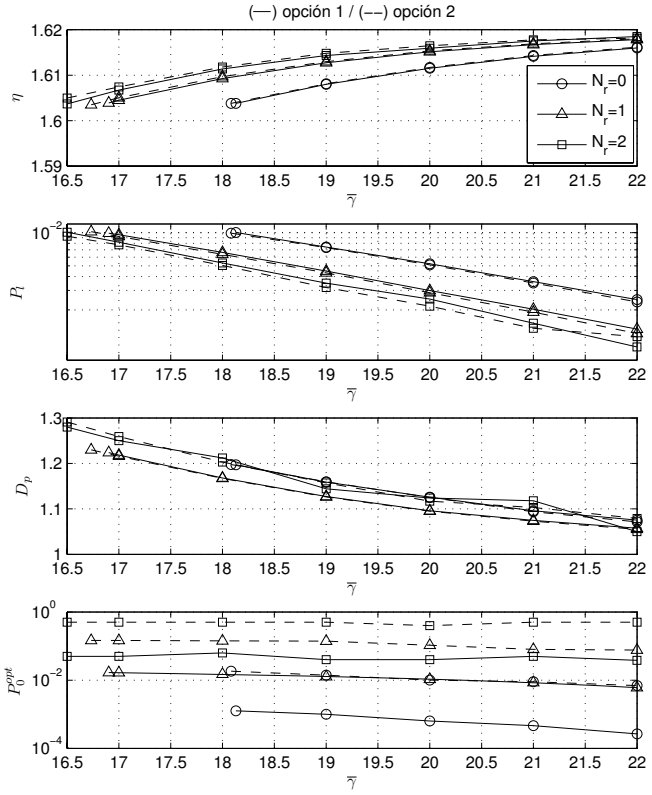


Figura 7. Optimización en función de $\overline{\gamma}$.

tramas) D_p y el valor de la PER objetivo óptima P_0^{opt} para diferentes valores de la tasa de generación de paquetes λ, y para las dos opciones consideradas en la determinación de los umbrales correspondientes a los modos de transmisión. Se hallan representadas las curvas correspondientes a un sistema en el que no se permiten las retransmisiones así como a dos sistemas en los que se utiliza un protocolo ARQ truncado con un número máximo permitido de retransmisiones igual a 1 y 2, respectivamente. Como puede observarse, para valores crecientes de λ las pérdidas de paquetes aumentan para un valor dado de N_r, hasta que P_l sobrepasa el valor máximo aceptable, haciéndose entonces necesario incrementar el número de retransmisiones permitidas N_r. Se aprecia claramente como el valor de P_0^{opt} se va haciendo más grande a medida que aumenta λ, ya que a medida que se incrementa la tasa de llegada de paquetes será necesario utilizar TMs de mayor orden para poder reducir la tasa de pérdida de paquetes por desbordamiento del buffer y maximizar así el throughput. El valor de D_p se mantiene en todo momento por debajo del máximo fijado, y se observa que un incremento de N_r no necesariamente implica un mayor retardo, puesto que a medida que aumenta el número permitido de retransmisiones no sólo se incrementa el throughput, sino también el tamaño medio de la cola, con lo que D_p puede incluso reducirse, tal como se observa en el caso del cambio de $N_r = 0$ a $N_r = 1$. Cabe destacar una vez más la similitud en el comportamiento de las dos opciones consideradas para el esquema AMC, donde la única diferencia consiste fundamentalmente en el valor de P_0^{opt} que, tal como hemos explicado anteriormente, es mayor para la *opción 2*.

Con el objetivo de analizar el proceso de optimización en función de la SNR media se ha extendido el estudio del sistema anterior para el caso en el que se considera un D-BMAP caracterizado por la matriz de probabilidades de transición $\boldsymbol{U}_{t1}$. En la Figura 7 se hallan representados los valores de η, P_l, D_p y P_0^{opt} para diferentes valores de $\overline{\gamma}$, y para las dos opciones consideradas en la determinación de los umbrales correspondientes a los modos de transmisión. Los gráficos muestran como a medida que se reduce la SNR media recibida las pérdidas de paquetes se van incrementando hasta llegar a superar el valor de $P_{l\,\text{máx}}$, lo que requiere el aumento del valor de N_r. De la misma manera que en los resultados reflejados en la Figura 6, se observa como el comportamiento de la *opción 1* y de la *opción 2* son análogos, distinguiéndose únicamente por los valores de P_0^{opt}, que tal y como ya se ha comentado, son mayores en la segunda opción.

VI. Conclusiones

En este artículo hemos presentado un novedoso marco teórico para el diseño intercapas en sistemas inalámbricos con un esquema AMC en la capa física y un protocolo ARQ truncado en la capa de enlace de datos, basado en el uso de una cadena de Markov bidimensional de primer orden para el modelado del canal inalámbrico y un modelo de colas a nivel de enlace que generaliza las herramientas analíticas propuestas por Wang *et al.* [11] y Le *et al.* [12]. A partir de ellos, se han obtenido las expresiones analíticas correspondientes a los parámetros fundamentales de análisis del comportamiento del sistema, esto es, throughput, retardo medio de paquete y tasa media de pérdida de paquetes tanto por desbordamiento del

buffer como por exceder el número máximo permitido de retransmisiones. El modelo analítico desarrollado se utiliza entonces para formular un diseño intercapas concebido como un problema de optimización con restricciones que permite analizar de manera conjunta el impacto sobre los parámetros de medida de QoS tanto del esquema AMC en la capa física como del protocolo ARQ en la capa de enlace de datos. Se han utilizado numerosos ejemplos numéricos con el objetivo de ilustrar el comportamiento del sistema con las dos opciones propuestas para la determinación del esquema AMC, así como el diseño intercapas propuesto.

Este marco teórico puede aplicarse en cualquiera de la multitud de sistemas actuales que combinan la utilización de un esquema AMC en la capa física con un protocolo ARQ en la capa de enlace de datos, como pueden ser el estándar IEEE 802.11 [20] o bien el estándar IEEE 802.16 [24]. En estos sistemas puede llevarse a cabo un proceso de optimización del throughput medio, asegurando al mismo tiempo el cumplimiento de los requisitos de QoS mediante la aplicación del diseño intercapas propuesto en el presente trabajo.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el Ministerio de Educación y Ciencia y el FEDER dentro del proyecto COSMOS (TEC2008-02422) y por el Govern de les Illes Balears a través del proyecto PCTIB-2005GC1-09.

REFERENCIAS

[1] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, "Cross-layer design for wireless networks," *IEEE Commun. Magazine*, vol. 41, pp. 74–80, Oct. 2003.

[2] V. Srivastana and M. Motani, "Cross-layer design: a survey and the road ahead," *IEEE Commun. Magazine*, vol. 43, pp. 112–119, Dec. 2005.

[3] V. Kawadia and P. R. Kumar, "A cautionary perspective on cross-layer design," *IEEE Wireless Commun.*, pp. 3–11, Feb. 2005.

[4] Q. Liu, S. Zhou, and G. B. Giannakis, "Cross-layer combining of adaptive modulation and coding with truncated ARQ over wireless links," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1746–1755, Sept. 2004.

[5] ——, "Queuing with adaptive modulation and coding over wireless links: cross-layer analysis and design," *IEEE Trans. Wireless Commun.*, vol. 4, no. 3, pp. 1142–1153, May 2005.

[6] ——, "Cross-layer scheduling with prescribed QoS guarantees in adaptive wireless networks," *IEEE Journal on Selected Areas in Commun.*, vol. 23, no. 5, pp. 1056–1066, May 2005.

[7] L. B. Le, E. Hossain, and A. S. Alfa, "Service differentiation in multirate wireless networks with weighted round-robin scheduling and ARQ-based error control," *IEEE Trans. Commun.*, vol. 54, no. 2, pp. 208–215, Feb. 2006.

[8] ——, "Radio link level performance evaluation in wireless networks using multi-rate transmission with ARQ-based error control," *IEEE Trans. Wireless Commun.*, vol. 5, no. 10, pp. 2647–2653, Oct. 2006.

[9] F. Ishizaki and G. U. Hwang, "Cross-layer design and analysis of wireless networks using the effective bandwidth function," *IEEE Trans. Wireless Commun.*, vol. 6, no. 9, pp. 3214–3219, Sept. 2007.

[10] M. Poggioni, L. Rugini, and P. Banelli, "Analyzing performance of multi-user scheduling jointly with AMC and ARQ," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, Nov. 2007, pp. 3483–3488.

[11] X. Wang, Q. Liu, and G. B. Giannakis, "Analizing and optimizing adaptive modulation coding jointly with ARQ for QoS-guaranteed traffic," *IEEE Trans. Veh. Technol.*, vol. 56, no. 2, pp. 710–720, March 2007.

[12] L. B. Le, E. Hossain, and T. Le-Ngoc, "Interaction between radio link level truncated ARQ, and TCP in multi-rate wireless networks: a cross-layer performance analysis," *IET Communications*, vol. 1, no. 5, pp. 821–830, 2007.

[13] C. C. Tan and N. C. Beaulieu, "On first-order Markov modeling for the Rayleigh fading channel," *IEEE Trans. Commun.*, vol. 48, no. 12, pp. 2032–2040, 2000.

[14] R. H. Clarke, "A statistical theory of mobile radio reception," *Bell System Tech. Journal*, vol. 47, no. 6, pp. 957–1000, Sept. 1968.

[15] J. Ramis, L. Carrasco, and G. Femenias, "Cross-layer design of multi-rate wireless systems using AMC with ARQ-based error control. a two dimensional Markov model approach," in *Proc JITEL*, Set. 2008, pp. 1–8.

[16] ——, "A two-dimensional Markov model for cross-layer design in AMC/ARQ-based wireless networks," in *Proc IEEE GLOBECOM*, Dec. 2008, pp. 4637–4642.

[17] E. Biglieri, G. Caire, and G. Taricco, "Limiting performance of block fading channels with multiple antennas," *IEEE Trans. Inf. Theory*, vol. 47, no. 4, pp. 1273–1289, May 2001.

[18] J. G. Kim and M. M. Krunz, "Delay analysis of selective repeat ARQ for a Markovian source over wireless channel," *IEEE Trans. Veh. Technol.*, vol. 49, no. 5, pp. 1968–1981, Sept. 2000.

[19] C. Blondia, "A discrete time batch markovian arrival process as b-isdn traffic model," *Belgian Journal of Operations Research, Statistics and Computer Science*, vol. 32, no. 3/4, pp. 3–23, 1993.

[20] IEEE, *802.11: Standard for Wireless LAN Medium Access Control and Physical Layer Specifications*. New York: IEEE, 1997.

[21] J. Max, "Quantization for minimum distortion," *IRE Trans. Information Theory*, vol. IT-6, pp. 7–12, March 1960.

[22] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Information Theory*, vol. IT-28, pp. 129–137, March 1982.

[23] L. Kleinrock, *Queuing Systems*. New York: Wiley, 1975, vol. I.

[24] IEEE, *802.16: Standard for Local and metropolitan area networks*. New York: IEEE, 2004.

Análisis y Optimización del Control de Flujo en HSDPA

Gaspar Pedreño, Juan J. Alcaraz y Fernando Cerdán
Departamento de Tecnologías de la Información y las Comunicaciones,
Universidad Politécnica de Cartagena,
Plaza del Cronista Isidoro Valverde, Edificio "La Milagrosa", CP. 30202 Cartagena.
{gaspar.pedreno, juan.alcaraz, fernando.cerdan}@upct.es

***Resumen*—Con la incorporación de HSDPA en las redes UMTS la función de *scheduling* se ha desplazado desde la RNC hasta el Nodo B, con la consiguiente necesidad de unos nuevos buffers en el Nodo B. A su vez, esta nueva distribución de la capacidad de almacenamiento entre la RNC y el Nodo B requiere de un mecanismo de control de flujo que regule la transferencia de datos entre ambos. Este artículo presenta un detallado estudio analítico donde este control de flujo es abordado como un problema de optimización cuadrática. Además, se propone un nuevo esquema de control de flujo que minimiza el retardo extremo a extremo gracias a que el nuevo algoritmo tiene en cuenta también la ocupación de los buffers de la RNC que hasta ahora no se había considerado en algoritmos anteriores.**

***Palabras Clave*—HSDPA, Control de Flujo, Optimización, Programación Dinámica**

I. Introducción

A. HSDPA

La evolución del mercado de la telefonía móvil ha supuesto una fuerte demanda de sistemas de mayor capacidad así como de tasas de transferencia más altas. En este contexto, el 3GPP (*3rd Generation Partnership Project*) extendió la especificación de WCDMA (*Wideband Code Division Multiple Access*) en la *Release 5* con el concepto HSDPA (*High Speed Downlink Packet Access*) [1]. HSDPA, diseñado específicamente para tráfico a ráfagas (servicios de tipo *interactive*, *streaming* y *background*), tiene como principales objetivos aumentar la velocidad de transmisión de datos en sentido *downlink*, mejorar la QoS percibida por el usuario y lograr un menor coste por bit entregado.

El concepto HSDPA se basa en la implementación de un nuevo canal de transporte compartido en sentido *downlink*, denominado HS-DSCH *(High Speed Downlink Shared Channel)* [1], y en la transferencia de algunas funcionalidades de la capa MAC desde la RNC *(Radio Network Controller)* hasta el Nodo B. HSDPA presenta múltiples novedades con respecto a WCDMA, entre las que destacan la codificación y modulación adaptativas (*fast link adaptation*), Hybrid-ARQ (HARQ), planificación rápida de paquetes (*fast packet scheduling*) y una reducción del intervalo de tiempo de transmisión (TTI) a 2 ms [2]. Estas características son soportadas en una nueva sub-capa MAC que se introduce en el Nodo B y que se conoce como MAC-hs (*Medium Access Control-high speed*). El resto de capas/protocolos de la red de acceso radio que se encuentran por encima de la capa MAC son los mismos que en la arquitectura que se presentó en *Release 99* para UMTS, sin modificación alguna. No obstante, el protocolo RLC (*Radio Link Control*) sólo puede operar en modo con o sin reconocimiento, pero no en modo transparente debido al cifrado. La Fig. 1 ilustra la estructura de protocolos de la interfaz radio para una red UMTS con funcionalidad HSDPA. En comparación con la arquitectura de *Release 99*, el desplazamiento del planificador de paquetes (*scheduler*) desde la RNC hasta el Nodo B es el cambio más notable. La motivación de este cambio es la necesidad de información reciente sobre el estado del canal radio por parte de los mecanismos de adaptación al canal (*link adaptation*) y del propio *scheduler*, a fin de poder realizar un seguimiento instantáneo de las condiciones radio de cada usuario. Ahora, al tomarse las decisiones de *scheduling* en la capa MAC-hs del Nodo B, se requieren nuevos buffers en el Nodo B para almacenar los datos de cada usuario que esperan ser transmitidos a través de la interfaz radio. Estos buffers se denominan colas de prioridad y cada uno recibe información de un solo flujo de datos desde la RNC. Puede haber hasta 8 colas de prioridad para cada usuario (*User Equipment, UE*) [1].

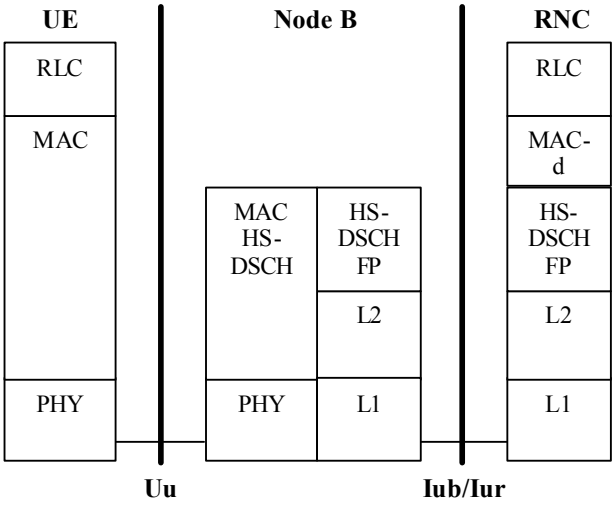


Fig. 1. Estructura de protocolos para la interfaz radio de HSDPA

B. Motivación

La distribución de la capacidad de almacenamiento entre la RNC y el Nodo B implica la necesidad de un mecanismo de control de flujo que regule la transferencia de datos (tramas RLC) desde la RNC al Nodo B. El objetivo de este mecanismo es mantener los buffers del Nodo B a un nivel tal que, por un lado, la capacidad del canal radio no sea

malgastada y, por otro, el retardo de encolamiento en el Nodo B no sea demasiado alto. Es decir, el buffer debe estar lo suficientemente lleno como para que nunca falten datos a la hora de transmitir por la interfaz radio. Por otro lado, dichos buffers no deben estar demasiado llenos ya que, en caso de producirse un handover (cambio de celda/Nodo B por parte del terminal móvil), los datos que estaban almacenados en el Nodo B inicial son eliminados y la RNC debe volver a transmitirlos hacia el nuevo Nodo B que controle ahora al usuario.

El mecanismo de control de flujo recogido en las especificaciones del 3GPP para el canal HS-DSCH [3] es el mismo que se propuso para los canales dedicados (*Dedicated Channels*, DCH) en la *Release '99* y se conoce como un sistema basado en créditos. Sin embargo, las especificaciones básicamente se limitan a definir los formatos de las tramas de señalización *HS-DSCH Capacity Request* y *HS-DSCH Capacity Allocation*, y de datos *HS-DSCH Data Frames* que se utilizan durante el proceso [3]. Por último, también se especifica que dicho control de flujo se debe realizar de forma independiente para cada flujo de datos con el fin de proporcionar un trato equitativo a todos los flujos.

Partiendo de estas premisas son varios los controles de flujo para HSDPA aparecidos durante los últimos años en la literatura científica. Un esquema que se utiliza habitualmente consiste en observar y controlar directamente el nivel del buffer en el Nodo B para cada flujo de datos de forma que el tiempo de encolamiento no supere un valor previamente definido [4], [5]. Este esquema será descrito y analizado en detalle en la siguiente sección desde un punto de vista de optimización matemática. En [6] los autores proponen un algoritmo de asignación de recursos para la interfaz Iub basado en una estimación del throughput de HSDPA a través de la interfaz aire mediante una cadena de Markov. Otro esquema de control de flujo para HSDPA es mostrado en [7]. Este esquema previene el desbordamiento de los buffers del Node B fijando un parámetro de control que determina el nivel máximo de ocupación en dichos buffers.

En este artículo se realiza un estudio analítico del control de flujo de HSDPA. Dicho control se plantea como un problema de optimización cuadrática que se resuelve mediante técnicas de programación dinámica (*Dynamic Programming*, DP) [8]. Además, se propone un nuevo algoritmo de control de flujo basado en estas mismas técnicas que minimiza el retardo extremo a extremo. Esto se consigue gracias a que el nuevo algoritmo tiene en cuenta también la ocupación de los buffers de la RNC que hasta ahora no se había considerado en algoritmos anteriores. Por último, dicho algoritmos es puesto a prueba en distintas situaciones mediante simulación computacional.

El resto del artículo está organizado de la siguiente forma. En la sección 2 se describirá el modelo matemático utilizado para caracterizar el sistema de control de flujo de HSDPA y se resolverá el problema de optimización derivado de dicho modelo matemático. Partiendo del estudio anterior, en la tercera sección se presentará un nuevo algoritmo que consigue minimizar el retardo extremo a extremo. En la sección 4, se muestran los resultados obtenidos mediante simulación para ambos algoritmos. Por último, la sección 5 recoge las conclusiones y líneas futuras de este trabajo.

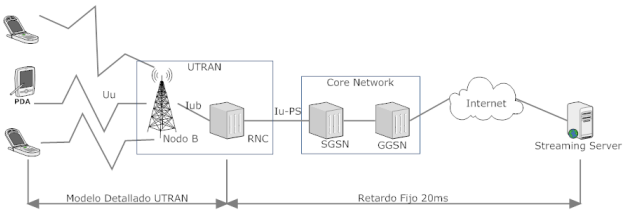


Fig. 2. Arquitectura de la red HSDPA considerada

II. Control de Flujo en HSDPA

A. Modelo del sistema

El escenario considerado para realizar nuestro estudio es una celda 3G con funcionalidad HSDPA (ver Fig. 2), donde varios usuarios (*User Equipments*, UEs) se conectan al Nodo B a través del canal *High Speed Downlink Shared Channel* (HS-DSCH) en sentido *downlink* y mediante un canal dedicado (DCH) en sentido *uplink*. La interfaz Iub, la cual conecta el Nodo B y la RNC, se supone de capacidad constante para el tráfico HSDPA. Asimismo, se considera que la RNC se conecta directamente a Internet a través de un enlace de capacidad constante. En una situación real, la capacidad de la Iub disponible para HSDPA es variable y depende del tráfico de los canales dedicados pero este hecho provoca picos de retardo y retardos impredecibles que dificultaría aislar la influencia de este efecto en las prestaciones de los distintos esquemas de control de flujo. El procedimiento encargado de controlar estos cambios en la capacidad de la Iub es el control de congestión (fuera del ámbito de estudio de este artículo). En futuras investigaciones incorporaremos dicho mecanismo a nuestro análisis a fin de estudiar las interacciones entre ambos controles (flujo y congestión).

Por simplicidad consideraremos sólo un flujo MAC-d por usuario, es decir, una sóla conexión de datos por usuario. Cada uno de estos flujos está asociado a una cola de prioridad del Nodo B, que guarda los datos que están a punto de ser transmitidos por la interfaz radio, y a un buffer en la RNC donde llega la información de dicho flujo desde Internet. Por tanto, la función del control de flujo consistirá en determinar la cantidad de información a transmitir para cada conexión de datos desde la RNC al Nodo B en un periodo de tiempo de longitud fija (cada conexión de datos tiene su propio proceso de control de flujo independiente del resto, tal y como describe el 3GPP en [3]). Dicho periodo es lo que comúnmente llamamos time-slot. Con el fin de obtener matemáticamente una política que determine las decisiones del control de flujo de HSDPA, cada conexión de datos entre la RNC y el Nodo B se modela como un sistema lineal en tiempo discreto. Cada time-slot se corresponde con la unidad de tiempo mínima en nuestro sistema en tiempo discreto, T_u.

Denotaremos por $q_i[n]$ la longitud en tramas RLC de la cola del Nodo B para el usuario *i* al inicio del time-slot n, y por $r_i[n]$ la tasa de transferencia de tramas RLC que son transmitidas por primera vez a través del canal radio para el usuario *i* durante el time-slot n (tasa efectiva). Sea $v_i[n]$ el número de *créditos*/tramas RLC concedidas por la RNC al

usuario *i* para el time-slot n. La evolución de la cola en el Nodo B para cada usuario *i* viene dada por la ecuación lineal:

$$q_i[n+1] = q_i[n] + v_i[n] - r_i[n] \cdot T_u \quad (1)$$

Dada la variabilidad de la calidad del canal radio, la tasa de transmisión efectiva $r_i[n]$ puede cambiar de manera impredecible siendo desconocido su valor al inicio del time-slot n. En nuestro modelo matemático la representaremos como:

$$r_i[n] = \overline{r}_i[n] + \xi_i[n] \quad (2)$$

dónde $\overline{r}_i[n]$ es el valor esperado de $r_i[n]$ y $\xi_i[n]$ es una variable aleatoria de media cero y varianza finita que consideraremos como un error en la estimación de $r_i[n]$.

B. Análisis Matemático

Genéricamente, la mayor parte de algoritmos diseñados para llevar a cabo el control de flujo en HSDPA regulan la transferencia de datos desde la RNC hasta el Nodo B con el objetivo de conseguir un tiempo de encolamiento predefinido T_w para los buffer del Nodo B $q_i[n]$ [4], [5]. En consecuencia, el control de flujo trata de mantener el nivel del buffer para cada flujo a un valor objetivo $Q_i[n]$:

$$Q_i[n] = r_i[n] \cdot T_w[n] \quad (3)$$

Para cada usuario *i*, cada time-slot el control de flujo trata de compensar la diferencia entre el nivel deseado $Q_i[n]$ y el nivel real $q_i[n]$. Hasta la fecha, los artículos que abordan este problema muestran directamente las expresiones que lo resuelven pero no explican de donde vienen dichos resultados. Sin embargo, en este artículo se formula el problema de control de flujo en HSDPA como un problema de optimización cuadrática. Así, la función objetivo a minimizar para cada usuario es:

$$E\{\textstyle\sum_{n=0}^{K}[(q_i[n] - Q_i[n])^2 + (v_i[n] - r_i[n] * T_u)^2]\} \quad (4)$$

dónde el primer término representa la penalización por desviarse del valor objetivo de la cola del Nodo B y el segundo término es una medida de la calidad con la que la tasa de transferencia de la RNC (*créditos*) sigue la tasa efectiva del canal. K es el número de etapas o time-slots sobre los que se minimiza.

Introduciendo los siguientes cambios de variable:

$$\begin{array}{rcl} x_i[n] & := & q_i[n] - Q_i[n] \\ u_i[n] & := & v_i[n] - \overline{r}_i[n] * T_u \end{array} \quad (5)$$

y usando notación vectorial

$$\begin{array}{rcl} \mathbf{x}_n & = & (x_1[n], \ldots, x_M[n])' \\ \mathbf{u}_n & = & (u_1[n], \ldots, u_M[n])' \\ \mathbf{w}_n & = & (\xi_1[n] \cdot T_u, ..., \xi_M[n] \cdot T_u)' \end{array} \quad (6)$$

la ecuación del sistema (1) con M usuarios en la celda se puede expresar como:

$$\mathbf{x}_{n+1} = \mathbf{A}_n\mathbf{x}_n + \mathbf{B}_n\mathbf{u}_n + \mathbf{w}_n = \mathbf{x}_n + \mathbf{u}_n + \mathbf{w}_n \quad (7)$$

donde $\mathbf{x}_n$ es el vector de estados, $\mathbf{u}_n$ es el vector de control y $\mathbf{w}_n$ es el vector de perturbaciones. En nuestro caso, tanto $\mathbf{A}_n$ como $\mathbf{B}_n$ son matrices identidad de orden $M \times M$.

Por su parte la función de coste o función objetivo (4) se puede expresar ahora de la siguiente forma:

$$\begin{array}{c} E\left\{\mathbf{x}'_K\mathbf{Q}_K\mathbf{x}_K + \sum_{n=0}^{K-1}\left(\mathbf{x}'_n\mathbf{Q}_n\mathbf{x}_n + \mathbf{u}'_n\mathbf{R}_n\mathbf{u}_n\right)\right\} = \\ E\left\{\mathbf{x}_K{}^2 + \sum_{n=0}^{K-1}\left(\mathbf{x}_n{}^2 + \mathbf{u}_n{}^2\right)\right\} \end{array} \quad (8)$$

donde las matrices $\mathbf{Q}_n$ y $\mathbf{R}_n$ son matrices identidad de orden $M \times M$. Las primeras representan el coste asociado a la ocupación de los buffers y las segundas el coste asociado a los vectores de control. K es el número de etapas sobre las que se desea optimizar el sistema. En nuestro caso, estamos interesados en reducir los niveles de los buffers al final de cada time-slot y, por tanto, minimizaremos sobre una sola etapa.

La minimización de la función de coste se llevará a cabo mediante programación dinámica aproximada. Básicamente lo que haremos es ir calculando etapa a etapa cuál es la combinación de controles que consigue minimizar el estado siguiente (nivel buffers), empezando en orden inverso desde la última etapa (n+1 en nuestro caso) hasta llegar a la etapa inicial. Así, en la etapa n, nuestro objetivo será minimizar el coste del control $\mathbf{u}_n$ y del estado $\mathbf{x}_{n+1}$. El coste de la última etapa sería:

$$J_{n+1}(\mathbf{x}_{n+1}) = \mathbf{x}_{n+1}{}^2 \quad (9)$$

El objetivo es encontrar el vector de control $\mathbf{u}_n$ con el que se obtenga el mínimo coste desde la etapa n hasta la $n+1$. En la etapa n, el coste vendría dado por:

$$J_n(\mathbf{x}_n) = \min_{\mathbf{u}_n} E\left\{\mathbf{x}_n{}^2 + \mathbf{u}_n{}^2 + J_{n+1}(\mathbf{x}_{n+1})\right\} \quad (10)$$

aplicando (9) y sustituyendo $\mathbf{x}_{n+1}$ por su valor en (7) obtenemos:

$$\begin{array}{l} J_n(\mathbf{x}_n) = 2\mathbf{x}_n{}^2 + E\left\{\mathbf{w}_n{}^2 + 2\mathbf{x}'_n\mathbf{w}_n\right\} \\ +\min_{\mathbf{u}_n}\left\{2\mathbf{x}'_n\mathbf{u}_n + 2\mathbf{u}_n{}^2 + E\left\{2\mathbf{w}'_n\mathbf{u}_n\right\}\right\} \end{array} \quad (11)$$

Teniendo en cuenta que la esperanza de $\mathbf{w}_n$ es cero (sus componentes $\xi_i[n]$ son variables aleatorias de media cero) el último término de la expresión anterior desaparece. El vector de control que minimiza $J_n(\mathbf{x}_n)$ puede obtenerse directamente derivando (11) con respecto a $\mathbf{u}_n$ e igualando el resultado a cero, obteniendo la siguiente ley de control:

$$\mathbf{u}_n = -\frac{1}{2}\mathbf{x}_n \quad (12)$$

Deshaciendo los cambios de variable realizados en (5), obtenemos el número de créditos óptimo que debe enviar la RNC para cada flujo i:

$$v_i[n] = -\tfrac{1}{2}\left(q_i[n] - Q_i[n]\right) + r_i[n] \cdot T_u \quad (13)$$

Curiosamente, en [4] se propone esta misma expresión desde un planteamiento intuitivo.

III. ALGORITMO PROPUESTO

Con el objetivo de mejorar las prestaciones del algoritmo anterior, proponemos un nuevo esquema de control de flujo que no sólo tiene en cuenta el estado de los buffers del Nodo B a la hora de tomar sus decisiones sino también el de los buffers de la RNC. Con este nuevo algoritmo se consigue minimizar el retardo extremo a extremo de cada conexión de datos, pero con la contrapartida de aumentar ligeramente la ocupación de los buffers del Nodo B.

Denotaremos por $y_i[n]$ la ocupación del buffer de la RNC para el usuario *i* al inicio del time-slot n, y $\lambda_i[n]$ la tasa de llegada de tramas RLC para el usuario *i* a la RNC durante el time-slot n. La evolución de las colas en el Nodo B y la RNC para cada usuario *i* viene dado por el siguiente sistema de ecuaciones lineales:

$$\begin{array}{rcl} q_i[n+1] & = & q_i[n] + v_i[n] - r_i[n] \cdot T_u \\ y_i[n+1] & = & y_i[n] - v_i[n] + \lambda_i[n] \cdot T_u \end{array} \quad (14)$$

Tanto $r_i[n]$ como $\lambda_i[n]$ son variables aleatorias cuyo valor es desconocido al inicio del time-slot n y, por tanto, tenemos que aproximarlas por sus valores esperados, $\bar{r}_i[n]$ y $\bar{\lambda}_i[n]$ respectivamente. El sistema se reescribe ahora como:

$$\begin{array}{rcl} q_i[n+1] & = & q_i[n] + v_i[n] - (\bar{r}_i[n] + \xi_i[n]) \cdot T_u \\ y_i[n+1] & = & y_i[n] - v_i[n] + (\bar{\lambda}_i[n] + \delta_i[n]) \cdot T_u \end{array} \quad (15)$$

dónde ξ_i y δ_i son variables aleatorias de media cero que representan los errores asociados a las predicciones de r_i y λ_i respectivamente.

Al igual que en el anterior algoritmo, introducimos el cambio de variable $x_i[n] := q_i[n] - Q_i[n]$ siendo $Q_i[n] = r_i[n] \cdot T_w[n]$ el valor objetivo para la longitud del buffer (T_w tiempo de encolamiento predefinido). Con este cambio el nuevo sistema queda de la siguiente forma:

$$\begin{array}{rcl} x_i[n+1] & = & x_i[n] + v_i[n] - (\bar{r}_i[n] + \xi_i[n]) \cdot T_u \\ y_i[n+1] & = & y_i[n] - v_i[n] + (\bar{\lambda}_i[n] + \delta_i[n]) \cdot T_u \end{array} \quad (16)$$

Por simplicidad expresaremos el sistema en notación matricial considerando M usuarios activos en la celda HSDPA. Así, definimos los siguientes vectores:

$$\begin{array}{rcl} \mathbf{z}_n & = & (x_1[n], ..., x_M[n], y_1[n], ..., y_M[n])' \\ \mathbf{v}_n & = & (v_1[n], ..., v_M[n], v_1[n], ..., v_M[n])' \\ \mathbf{w}_n & = & (-\bar{r}_1[n] - \xi_1[n], ..., -\bar{r}_M[n] - \xi_M[n], \\ & & \bar{\lambda}_1[n] + \delta_1[n], ..., \bar{\lambda}_M[n] + \delta_M[n])' \end{array} \quad (17)$$

donde $\mathbf{z}_n$ es el vector de estados, $\mathbf{v}_n$ es el vector de control y $\mathbf{w}_n$ es el vector de perturbaciones. El sistema lineal (16) puede expresarse ahora en notación matricial como:

$$\mathbf{z}_{n+1} = \mathbf{A}_n \mathbf{z}_n + \mathbf{B}_n \mathbf{v}_n + T_u \cdot \mathbf{w}_n \quad (18)$$

dónde $\mathbf{A}_n$ es una matriz identidad de dimensión $2M \times 2M$ y $\mathbf{B}_n$ tiene la siguiente forma:

$$B = \begin{pmatrix} I_N \\ -I_N \end{pmatrix} \quad (19)$$

con I_N matriz identidad de dimensión $M \times M$.

El objetivo de nuestro algoritmo es minimizar la ocupación total del sistema y, por tanto, minimizar el retardo extremo a extremo. Para ello formularemos este problema de minimización como un problema de programación dinámica [8], asociando una función de coste al sistema anteriormente descrito, la cual deberemos minimizar. Asumiendo un coste cuadrático, dicha función de coste presenta el siguiente aspecto:

$$E\left\{ \mathbf{z}'_K \mathbf{Q}_K \mathbf{z}_K + \sum_{n=0}^{K-1} (\mathbf{z}'_n \mathbf{Q}_n \mathbf{z}_n + \mathbf{v}'_n \mathbf{R}_n \mathbf{v}_n) \right\} \quad (20)$$

donde las matrices $\mathbf{Q}_n$ son simétricas semidefinidas positivas y las matrices $\mathbf{R}_n$ son simétricas definidas positivas. K es el número de etapas sobre las que se desea optimizar el sistema. Al igual que en el algoritmo anterior, nos interesa reducir los buffers al final de cada slot y, por tanto, minimizaremos sobre una sola etapa.

Como vemos, minimizar la función de coste supone minimizar dos factores cuadráticos: por un lado, la ocupación de los buffers del Nodo B y RNC ($\mathbf{z}_n$) y, por otro, los vectores de control o créditos ($\mathbf{v}_n$). El primer factor se corresponde con nuestro objetivo inicial mientras que el segundo impone una restricción sobre los créditos que evita que obtengamos como solución al problema la solución trivial, es decir, asignar infinitos recursos.

Las matrices $\mathbf{Q}_n$ representan el coste asociado a la longitud de los buffers y pueden ser expresadas como:

$$\mathbf{Q}_n = \begin{pmatrix} \mathbf{Q}_{00n} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{11n} \end{pmatrix} \quad (21)$$

con todas las submatrices de dimensión $M \times M$ y valor:

$$\begin{array}{rcl} \mathbf{Q}_{00n} & = & \alpha_x[n]\mathbf{I} \\ \mathbf{Q}_{11n} & = & \alpha_y[n]\mathbf{I} \end{array} \quad (22)$$

donde $\mathbf{I}$ es las matriz identidad. $\alpha_x[n]$ y $\alpha_y[n]$ son los factores de ponderación aplicados a los buffers de Nodo B y RNC respectivamente, es decir, el coste o penalización asociada a la ocupación de dichos buffers. El ajuste de estos pesos es fundamental para el correcto funcionamiento del sistema.

Por su parte, $\mathbf{R}_n$ es una matriz identidad de orden $2M \times 2M$ y puede interpretarse como el coste asociado a los vectores de control $\mathbf{v}_n$ (créditos).

En general, las matrices $\mathbf{Q}_n$ y $\mathbf{R}_n$ pueden depender del índice pero como vamos a minimizar etapa a etapa nos referiremos a ellas como $\mathbf{Q}$ y $\mathbf{R}$. En la etapa n, nuestro objetivo será minimizar el coste del control $\mathbf{v}_n$ y del estado $\mathbf{z}_{n+1}$. El coste de esta última etapa sería:

$$J_{n+1}(\mathbf{z}_{n+1}) = \mathbf{z}'_{n+1} \mathbf{Q} \mathbf{z}_{n+1} \quad (23)$$

El objetivo es encontrar el vector de control $\mathbf{v}_n$ con el que se obtenga el mínimo coste desde la etapa n hasta la última etapa ($n+1$ en nuestro caso). En la etapa n, el coste vendría dado por:

$$J_n(\mathbf{z}_n) = \min_{\mathbf{v}_n} E\left\{ \mathbf{z}'_n \mathbf{Q} \mathbf{z}_n + \mathbf{v}'_n \mathbf{R} \mathbf{v}_n + J_{n+1}(\mathbf{z}_{n+1}) \right\} \quad (24)$$

aplicando (23) y sustituyendo $\mathbf{z}_{n+1}$ por su valor en (18) obtenemos:

$$\begin{array}{l} J_n(\mathbf{z}_n) = \mathbf{z}'_n (\mathbf{A}'\mathbf{Q}\mathbf{A} + \mathbf{Q}) \mathbf{z}_n + E\left\{ \mathbf{w}'_n \mathbf{Q} \mathbf{w}_n + 2\mathbf{z}'_n \mathbf{A}' \mathbf{Q} \mathbf{w}_n \right\} \\ + \min_{\mathbf{v}_n} \left\{ 2\mathbf{z}'_n \mathbf{A}' \mathbf{Q} \mathbf{B} \mathbf{v}_n + \mathbf{v}'_n (\mathbf{B}'\mathbf{Q}\mathbf{B} + \mathbf{R}) \mathbf{v}_n + E\left\{ 2\mathbf{w}'_n \mathbf{Q} \mathbf{B} \mathbf{v}_n \right\} \right\} \end{array} \quad (25)$$

Teniendo en cuenta la linealidad del operador esperanza, el último término de la expresión anterior se puede escribir como $2\overline{\mathbf{w}}'_n \mathbf{Q} \mathbf{B} \mathbf{v}_n$, donde $\overline{\mathbf{w}}_n = (-\bar{r}_1[n], ..., \bar{r}_M[n], \bar{\lambda}_1[n], ..., \bar{\lambda}_M[n])$. El vector de control que minimiza $J_n(\mathbf{z}_n)$ puede obtenerse directamente derivando (25) con respecto a $\mathbf{v}_n$ e igualando el resultado a cero

$$(\mathbf{B}'\mathbf{Q}\mathbf{B} + \mathbf{R}) \mathbf{v}_n + (\mathbf{z}'_n \mathbf{A}' \mathbf{Q} \mathbf{B} + \overline{\mathbf{w}}'_n \mathbf{Q} \mathbf{B})' = 0 \quad (26)$$

Despejando el valor de $\mathbf{v}_n$, obtenemos que la ley de control resultante es:

$$\mathbf{v}_n = -\left(\mathbf{R} + \mathbf{B}'\mathbf{Q}\mathbf{B}\right)^{-1} \mathbf{B}'\mathbf{Q}\mathbf{A} \left(\mathbf{z}_n + \mathbf{A}^{-1} \bar{\mathbf{w}}_n\right) \quad (27)$$

Sustituyendo en (27) las matrices por sus valores correspondientes obtenemos que el control aplicado para cada flujo de datos *i* debería ser:

$$v_i[n] = \frac{-1}{\alpha_x[n]+\alpha_y[n]} \\ (\alpha_x[n]\,(x_i[n] - r_i[n]\cdot T_u) - \alpha_y[n]\,(y_i[n] + \lambda_i[n]\cdot T_u)) \quad (28)$$

y deshaciendo el cambio de variable:

$$v_i[n] = \frac{-\alpha_x[n]}{\alpha_x[n]+\alpha_y[n]}\,(q_i[n] - Q_i[n] - r_i[n]\cdot T_u) + \\ \frac{\alpha_y[n]}{\alpha_x[n]+\alpha_y[n]}\,(y_i[n] + \lambda_i[n]\cdot T_u) \quad (29)$$

Como vemos los créditos concedidos dependen de dos factores, el primero relacionado con el estado de los buffers del Nodo B $(q_i[n] - Q_i[n] - r_i[n]\cdot T_u)$ y el segundo $(y_i[n] + \lambda_i[n]\cdot T_u)$ con los de la RNC. La clave está en configurar apropiadamente $\alpha_x[n]$ y $\alpha_y[n]$. Al principio se utilizaron valores fijos para estos parámetros pero inmediatamente se observó la necesidad de una configuración dinámica que permitiera balancear adecuadamente los datos entre RNC y Nodo B dependiendo de las condiciones del canal radio de cada usuario. Así, tras probar diferentes configuraciones se optó por la siguiente:

$$\begin{array}{rcl} 0 < \frac{q_i[n]}{Q_i[n]} < 1 & \Rightarrow & \alpha_x[n] = 1, \alpha_y[n] = 1 \\ 1 \leq \frac{q_i[n]}{Q_i[n]} < 2 & \Rightarrow & \alpha_x[n] = \frac{q_i[n]}{Q_i[n]} + 1, \alpha_y[n] = 1 \\ \frac{q_i[n]}{Q_i[n]} \geq 2 & \Rightarrow & \alpha_x[n] = 1, \alpha_y[n] = 0 \end{array} \quad (30)$$

Si el nivel del buffer del Nodo B aún no ha llegado al nivel objetivo $(0 < \frac{q_i[n]}{Q_i[n]} < 1)$ se ponderan por igual ambos factores (0.5), tanto el que depende del Nodo B como el de la RNC. De esta forma, se incrementa la tasa de trasnmisión de la RNC para que el buffer del Nodo B alcance lo antes posible el valor objetivo. Una vez alcanzado/superado el nivel objetivo $(1 \leq \frac{q_i[n]}{Q_i[n]} < 2)$, se toma $\alpha_x[n]$ mayor que $\alpha_y[n]$ de forma que se reduzca la importancia del factor relativo a los buffers de la RNC para frenar el envío de créditos. En caso de superar el umbral de $2Q_i[n]$ se reduce drásticamente la tasa, dejando de considerar la ocupación de la RNC.

IV. Evaluación de Prestaciones

A. Modelo de Simulación

Para la evaluación de prestaciones de los algoritmos presentados se ha implementado un modelo detallado de celda HSDPA que incluye los protocolos RLC, MAC-d y MAC-hs. El simulador ha sido desarrollado en OMNeT++, herramienta de simulación en c++ orientada a eventos [9].

Para la implementación de la capa física (canal radio) se ha optado por un modelo basado en curvas BLER, el cual está completamente descrito en [10]. El modelo elegido es un *ITU-T Pedestrian A* con velocidad de 3 km/h. Los equipos de usuario (UE's) informan al Nodo B del estado del canal radio por medio del Channel Quality Indicator (CQI) con un retardo constante de 6ms [10]. Los formatos de transporte (TF) son elegidos en la capa MAC-hs a partir de estos informes de forma que el BLER sea inferior al 10%.

El Nodo B se conecta a la RNC a través de un enlace punto a punto de ancho de banda constante e igual a 6Mbps. De esta forma se evita que la interfaz Iub actúe de cuello de botella e interfiera en las prestaciones de los algoritmos a estudiar. Por otra parte, se supone que Internet y el *core network* introducen un retardo constante de 20 ms en cada dirección.

Todos los experimentos se realizan con tráfico streaming video. Dicho tráfico se modela mediante unas fuentes de tasa constante que transmiten a 312 Kbps con un tamaño de paquete de 576 bytes, similares a las utilizadas en [12]. En consecuencia, al tratarse de tráfico en tiempo real la capa RLC funciona en modo sin reconocimiento (UM). El número máximo de retransmisiones en la capa MAC-hs es de 3 y el tamaño de la ventana de MAC-hs se fija en 12. Se considera que los terminales móviles son de categoría 6, lo que implica que la máxima velocidad a la que pueden recibir datos desde el canal HSDPA (HS-DSCH) es 3.6 Mbps. No se implementa multiplexación en código. Los enlaces ascendentes DCHs operan a 64 Kbit/s. El tamaño de las tramas RLC se ajusta a su valor típico, 320bits.

El tiempo de espera deseado en los buffers del Nodo B para cada flujo de datos se establece en $T_w = 100ms$. El tiempo de actualización T_u tiene que ser lo suficientemente pequeño como para permitir al flujo de control seguir con precisión las variaciones del canal de radio pero a su vez tiene que ser lo suficientemente grande como para mantener la carga de señalización entre RNC y Nodo B a un nivel razonable. Teniendo en cuenta este compromiso se ha fijado $T_u = 100ms$.

B. Resultados

En primer lugar se ha realizado un experimento para comparar las prestaciones de los dos algoritmos expuestos en este artículo frente a la situación sin control de flujo bajo tres schedulers diferentes: *Round Robin*, *Maximum C/I* y *Proportional Fair*. El primero de ellos elige el usuario que debe transmitir de forma equitativa siguiendo un orden secuencial. Con *Maximum C/I*, en cada TTI trasmite siempre el usuario que experimenta las mejores condiciones radio (mayor ratio portadora a interferencia), siendo el esquema más injusto. Por último, *Proportional Fair* es una combinación de los dos anteriores pues, pretende ser equitativo pero tiene en cuenta también el estado del canal radio a la hora de seleccionar al usuario. En [13] se describe detalladamente este *scheduler*. Los resultados mostrados en la Fig. 3 corresponden al promedio de diez réplicas de cien segundos de duración, usando un intervalo de confianza del 95%. Se considera que hay doce usuarios activos en la celda.

Como es lógico, en ausencia de control de flujo se consiguen los mejores resultados en cuanto a retardo para cualquier *scheduler* puesto que los datos atraviesan la RNC sin espera alguna, siendo transmitidos directamente hacia el Nodo B. De la misma forma, obtiene los niveles de throughput más altos pues el Nodo B dispone 'siempre' de datos para transmitir por la interfaz radio. Sin embargo, esta sitación no es deseable puesto que, los buffers del Nodo B no son demasiado grandes con lo que podrían desbordarse y, además, en el caso de que ocurra un handover todos los datos que estuvieran almacenados en el Nodo B original serían eliminados y la RNC debe volver a transmitirlos hacia el nuevo Nodo B que controlase la celda donde se hubiera desplazado el usuario. En lo que a los controles de flujo se refiere, se observa como nuestro algoritmo consigue mejorar significativamente el retardo extremo a extremo con respecto

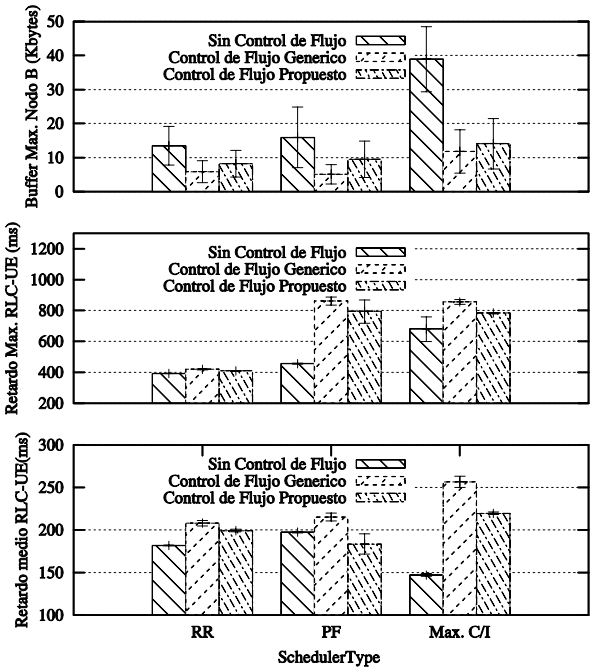


Fig. 3. Prestaciones frente a distintos schedulers

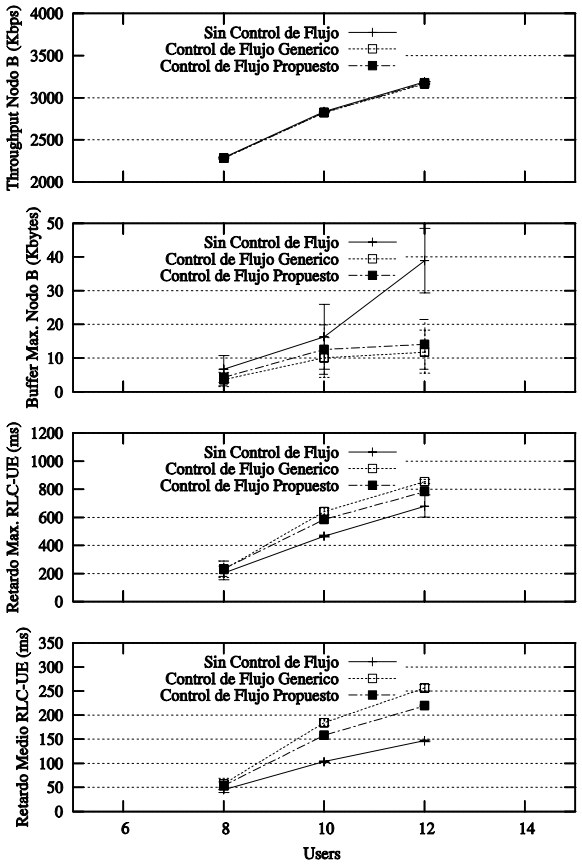


Fig. 4. Prestaciones en función del número de usuarios

al algoritmo genénico. Por contra, incrementa ligeramente el nivel del buffer del Nodo B.

Por último se comparan los algoritmos frente al número de usuarios activos en el sistema con un *scheduler Maximum C/I*. En la Fig. 4 se muestran los resultados de este experimento. Al igual que en la anterior gráfica, los datos corresponden al promedio de diez réplicas de cien segundos de duración usando un intervalo de confianza del 95%.

Como vemos, los resultados de este experimento están en línea con los anteriores. El algoritmo propuesto consigue mejores figuras de retardo (a costa de una mayor ocupación del buffer del Nodo B) que el algoritmo genérico. Por otro lado, cuánto mayor es el número de usuarios mayores diferencias se aprecian entre el rendimiento de los algoritmos. En el caso de los doce usuarios vemos como el throughput del Nodo B se acerca ya a los 3.6Mbit/s que, para usuarios de categoría 6, sería la tasa máxima a la que podría transmitir.

V. Conclusiones

En este artículo se ha abordado el problema del control de flujo de HSDPA desde una perspectiva diferente a los trabajos realizados hasta la fecha. Se ha planteado como un problema de optimización cuadrática para cuya resolución se ha hecho uso del método de programación dinámica. Se ha propuesto también un nuevo algoritmo de control de flujo que minimiza el retardo extremo a extremo desde la RNC hasta el UE.

En un futuro se pretende extender este análisis matemático al control de congestión del Iub con el fin de obtener un algoritmo capaz de desempeñar las dos funcionalidades: control de flujo y de congestión del Iub.

Agradecimientos

Este trabajo ha sido financiado por el proyecto TEC2007-67966-01/TCM (CON-PARTE-1) y desarrollado también en el marco del "Programa de Ayudas a Grupos de Excelencia de la Region de Murcia, Fundacion Seneca, Agencia de Ciencia y Tecnologia de la RM (Plan Regional de Ciencia y Tecnologia 2007/2010". Gaspar Pedreño agradece el apoyo del Ministerio de Educación y Ciencia a través de la beca FPU AP2006-01568.

Referencias

[1] *3GPP 25.308 "UTRA High Speed Downlink Packet Access (HSDPA);Overall description"*.
[2] *3GPP 25.950 "UTRA High Speed Downlink Packet Access"*.
[3] *3GPP 25.877 "HSDPA - Iub/Iur Protocol Aspects"*.
[4] Marc C. Necker and Andreas Weber, *Impact of Iub Flow Control on HSDPA System Performance*, 2nd International Symposium on Wireless Communication Systems, 2005.
[5] Szilveszter Nadas, Sandor Racz, Zoltan Nagy and Sandor Molnar, *Providing congestion control in the Iub Transport Network for HSDPA*, Globecom 2007.
[6] Xinzhi Yan, Jamil Y. Khan, and Brendan Jones, *An Adaptive Resource Management Technique for a HSDPA Network*, IFIP International Conference on Wireless and Optical Communications Networks, 2007.
[7] Hong Wei, Chen Shuping, Peng Mugen and Wang Wembo, *An efficient Iub flow control algorithm in HSDPA*, International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Commnication, 2007.
[8] Dimitri P. Bertsekas, *Dynamic Programing and Optimal Control, Vol. 1*, Prentice Hall, USA, 1987.
[9] Andres Vearga, *OMNeT++ 3.2 User Manual and API Reference*, www.omnetpp.org.
[10] N. Whillans (editor), *End-to-End Network Model for Enhanced UMTS. 1st SEACORN Project Deliverable D3.2v2*, 2003.

[11] Harri Holma and Jussi Reunanen, *3GPP Release 5 HSDPA Measurements*, International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2006.

[12] Marc C. Necker and Andreas Weber, *Protocol Interference Between Up- and Donlink Channels in HSDPA*, International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2006.

[13] Troels E. Kolding, *Link and System Performance Aspects of Proportional Fair Scheduling in WCDMA/HSDPA*, Vehicular Technology Conference, Vol. 2, pp. 492-499, 2003.

Evaluación comparativa de sistemas de comunicaciones con dos órbitas de reintentos

Mª Jose Domenech-Benlloch, Jose Manuel Gimenez-Guzman*,
Vicent Pla, Vicente Casares-Giner and Jorge Martinez-Bauset
Dept. Comunicaciones, Universidad Politécnica Valencia
Cami de Vera, s/n 46022, Valencia, Spain.
* Dept. Automática, Universidad de Alcalá
28871 Alcalá de Henares, Madrid, Spain
mdoben@doctor.upv.es, josem.gimenez@uah.es, {vpla,vcasares}@dcom.upv.es,jmartinez@upvnet.upv.es

***Resumen*—En las redes celulares que garantizan la movilidad de los usuarios mientras mantienen sus sesiones en curso, los reintentos ocurren como resultado del comportamiento de los usuarios y debido a los reintentos automáticos de los *handovers* bloqueados. Al modelar este tipo de redes obtenemos un sistema multiservidor con dos órbitas de reintentos. La complejidad analítica de este tipo de sistemas no permite obtener soluciones exactas de los principales parámetros de mérito, por lo que se debe recurrir a métodos aproximados. Todos los métodos aparecidos hasta la fecha se basan en el cálculo de las probabilidades de estado en régimen permanente. En este artículo se propone otra alternativa basada en las ecuaciones de Howard y se comparan sus prestaciones con los métodos más conocidos que han aparecido en la literatura. Los resultados muestran que esta solución supera ampliamente las propuestas previas en términos tanto de precisión como de coste computacional.**

***Palabras Clave*—Redes celulares, evaluación de prestaciones, sistemas con reintentos.**

I. Introducción

El fenómeno de los reintentos aparece en múltiples situaciones en las telecomunicaciones y en las redes de ordenadores. En este artículo centramos nuestra atención en una red de comunicaciones genérica que garantiza la movilidad sin interrupciones a sus clientes por medio de una arquitectura celular. En este tipo de redes, el área de cobertura se divide en células y los clientes pueden moverse a lo largo de las diferentes células de la red. Cuando un usuario con una comunicación activa se mueve de una célula a otra, se produce un denominado traspaso o *handover*. Hoy en día, quizás el ejemplo más extendido de este tipo de redes son las redes celulares telefónicas —2G y 3G— pero la perspectiva actual es que en un futuro cercano existirán un elevado número de tecnologías que encajen en esta categoría, por ejemplo, Mobile IP, IEEE 802.16e —WiMAX— o IEEE 802.20 —MBWA.

Este artículo se centra en el caso en el que los reintentos aparecen no únicamente cuando un usuario resulta bloqueado sino también cuando un *handover* se bloquea, como en GSM [1]. Hasta donde llega nuestro conocimiento, el único artículo que ha considerado el efecto de ambos tipos de reintentos simultáneamente en las prestaciones de la red es [2]. A partir de este momento, nos referiremos al primer tipo de reintentos como remarcados y al último como reintentos automáticos, mientras que usaremos el término reintento para referirnos a cualquiera de ellos. Los *handovers* bloqueados se reintentarán automáticamente hasta que se tenga éxito o el usuario salga del área de *handover*. En el primer caso la sesión continuará sin que el usuario note ningún tipo de interrupción mientras que en el segundo caso la sesión terminará abruptamente. Por otro lado, la persistencia de los remarcados depende de la paciencia de los usuarios y un abandono conduce a un fallo en el establecimiento de la sesión. Otra diferencia es que el número máximo de reintentos automáticos se puede configurar por la red mientras que los remarcados están afectados por la aleatoriedad del comportamiento humano. Por lo tanto, ambos tipos de reintentos tienen diferentes características y como consecuencia deben considerarse dos órbitas de reintentos por separado en el análisis del sistema.

El modelado de reintentos ha sido un tema sujeto a numerosas investigaciones por el impacto que pueden llegar a tener en las redes de comunicaciones. Típicamente, se distinguen dos bloques funcionales en los modelos que consideran reintentos: un bloque que acomoda a los servidores —y posiblemente una cola de espera— y un segundo bloque donde se encuentran los usuarios que están reintentando su acceso al sistema, al cual se le suele denominar órbita de reintentos. Para solucionar este tipo de sistemas es necesario hacer uso de métodos aproximados [3]. Estos métodos suelen agruparse en tres categorías: aproximaciones, métodos truncados finitos y métodos truncados generalizados [3]. Centraremos nuestra atención únicamente en las dos últimas categorías, puesto que son los métodos que mejores prestaciones ofrecen [4]. Los métodos truncados finitos reemplazan el espacio de estados infinito original por otro de tamaño finito, para el que se pueden calcular las probabilidades de estado en régimen permanente. Por otro lado, los métodos truncados generalizados sustituyen el espacio de estados infinito original por otro espacio de estados infinito pero para el que podemos calcular las probabilidades de estado en régimen permanente. Esta última categoría de métodos suele ofrecer las mejores prestaciones [4].

Todas las aproximaciones aparecidas en la literatura hasta la fecha se basan en la resolución numérica de las ecuaciones de Kolmogorov en régimen permanente de la cadena de Markov en tiempo continuo —CTMC— que describe el modelo estudiado. Sin embargo, muy recientemente, Leino et al. [5] han propuesto una aproximación alternativa para evaluar procesos de Markov con espacio de estados infinito. El nuevo método, denominado *Value Extrapolation* —VE—

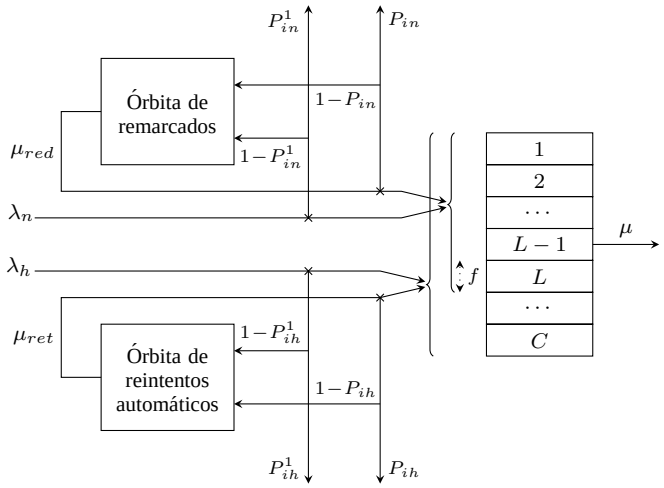


Fig. 1. Modelo del sistema.

no se basa en la resolución de las ecuaciones de balance globales, sino que considera el sistema en su forma de proceso de decisión de Markov —MDP— y resuelve los *relative state values* de las ecuaciones de Howard formuladas para un espacio de estado truncado.

El principal objetivo de este trabajo es el de aplicar el método VE a un sistema con dos órbitas de reintentos y comparar sus prestaciones con otros posibles métodos aproximados. Esta evaluación de prestaciones se realiza en un escenario de red celular que garantiza la movilidad de los usuarios como el descrito más arriba. De forma general, concluimos que VE supera al resto de métodos en un amplio rango de escenarios no sólo en términos de precisión sino también en términos de coste computacional, así que su uso es altamente recomendable.

El resto del artículo se estructura del siguiente modo. Primero, se describe la red celular bajo estudio y su modelo asociado. En la sección III, se enumeran y explican las principales características de los métodos con los que se compara VE. La sección IV está dedicada a la descripción de VE y cómo se ha aplicado al modelo bajo estudio. En la sección V se lleva a cabo un estudio numérico y finalmente en la sección VI se realiza un breve resumen y se destacan las principales conclusiones del artículo.

II. Descripción del sistema y modelado

Se considera una red móvil celular que emplea un esquema de asignación de recursos por célula fijo donde cada célula se sirve mediante una estación base diferente, donde C es el número de recursos en la célula. Como se muestra en la Fig. 1 hay dos flujos de llegadas: el primero representa a las sesiones nuevas y el segundo a los *handovers* que provienen de células adyacentes. Ambos procesos de llegada se consideran de Poisson con tasas λ_n and λ_h respectivamente. Esto nos conduce a una tasa total de llegadas al sistema de tasa $\lambda = \lambda_n + \lambda_h$. Por tratabilidad matemática, el tiempo de ocupación de canal se considera distribuido exponencialmente con tasa μ [6].

En general, el bloqueo de una petición de establecimiento de una nueva sesión se considera menos perjudicial que el bloqueo de un *handover*. Por lo tanto, se debe incluir una política de control de admisión que garantice esa priorización de *handovers* —y reintentos automáticos— sobre sesiones nuevas —y sus remarcados asociados— y, por lo tanto, asegure un cierto grado de calidad de servicio —QoS. La técnica más extendida consiste en reservar algunos recursos a los flujos de mayor prioridad, siendo en nuestro caso los *handovers* y sus reintentos automáticos asociados. Esta técnica puede generalizarse a una reserva fraccional, y se convierte en la política de control de admisión denominada *Fractional Guard Channel* [7] —FGC. La política FGC se caracteriza mediante un único parámetro t ($0 \leq t \leq C$). Las sesiones nuevas y sus remarcados se aceptan con probabilidad 1 cuando hay menos de $L = \lfloor t \rfloor$ recursos en uso y con probabilidad $f = t - L$ cuando hay exactamente L recursos en uso. Si hay más de L recursos ocupados, no se aceptan este tipo de peticiones. Los *handovers* y sus reintentos automáticos asociados sólo se rechazan cuando el sistema está completamente ocupado.

Cuando se bloquea una petición de establecimiento de nueva sesión, de acuerdo con la Fig. 1, esta sesión pasa a formar parte de la órbita de remarcados con probabilidad $(1 - P_{in}^1)$ o abandona el sistema con probabilidad P_{in}^1. Si un remarcado no tiene éxito, la sesión vuelve a la órbita de remarcados con probabilidad $(1 - P_{in})$, remarcando después de un tiempo distribuido exponencialmente con tasa μ_{red}. Los remarcados pueden acceder a los mismos recursos del sistema que las sesiones nuevas. Nótese que P_{in}^1 y P_{in} modelan el fenómeno de la impaciencia, es decir, la posibilidad de abandonar el sistema sin haber sido atendido. De forma similar, P_{ih}^1, P_{ih} y μ_{ret} son los parámetros análogos para los reintentos automáticos. Existen diferentes parámetros de prestaciones que generalmente se usan para describir el comportamiento de este tipo de sistemas con reintentos. Por un lado, las ampliamente utilizadas probabilidades de bloqueo de nuevas sesiones —P_b^n— y *handovers* —P_b^h. Por otro lado, el número medio de usuarios remarcando —N_{red}— y de *handovers* reintentando automáticamente —N_{ret}— pueden describir de una forma más precisa el fenómeno de los reintentos.

El modelo considerado puede representarse mediante un CTMC tridimensional (k, m, o), donde k representa el número de sesiones que están en los servidores, m especifica el número de sesiones en la órbita de remarcados y o representa el número de sesiones en la órbita de reintentos. El espacio de estados puede representarse mediante:

$$\mathcal{S} := \{(k, m, o) : k \leq C; m \in \mathbb{Z}_+; o \in \mathbb{Z}_+\}. \qquad (1)$$

Las tasas de transición de este modelo se representan en la Tabla I. Las principales características matemáticas consisten en tener dos dimensiones infinitas —el espacio de estados del modelo es $\{0, \ldots, C\} \times \mathbb{Z}_+ \times \mathbb{Z}_+$— y la heterogeneidad a lo largo de ellas. Esta heterogeneidad se produce por las tasas de reintentos automáticos y remarcados, que dependen del número de usuarios que tengan en sus respectivas órbitas.

III. Métodos de resolución

La teoría de colas clásica, véase por ejemplo [8], se ha desarrollado para paseos aleatorios en $\{0, \ldots, C\} \times \mathbb{Z}_+$ con transiciones sujetas a condiciones de homogeneidad. Cuando no existe tal homogeneidad el problema de calcular las probabilidades de estado no se ha desarrollado más allá de

Tabla I
TASAS DE TRANSICIÓN DEL MODELO EXACTO.

Transición	Condición	Tasa
$(k,m,o) \to (k+1,m,o)$	$0 \le k \le L-1$	λ
	$k = L$	$\lambda_h + f\lambda_n$
	$L < k < C$	λ_h
$(k,m,o) \to (k+1,m,o-1)$	$0 \le k \le C-1$	$o\mu_{ret}$
$(k,m,o) \to (k,m,o-1)$	$k = C$	$o\mu_{ret}P_{ih}$
$(k,m,o) \to (k+1,m-1,o)$	$0 \le k \le L-1$	$m\mu_{red}$
	$k = L$	$m\mu_{red}f$
$(k,m,o) \to (k,m-1,o)$	$k = L$	$m\mu_{red}(1-f)P_{in}$
	$L < k \le C$	$m\mu_{red}P_{in}$
$(k,m,o) \to (k-1,m,o)$	$1 \le k \le C$	$k\mu$
$(k,m,o) \to (k,m,o+1)$	$k = C$	$\lambda_h(1-P_{ih}^1)$
$(k,m,o) \to (k,m+1,o)$	$k = L$	$\lambda_n(1-P_{in}^1)(1-f)$
	$L < k \le C$	$\lambda_n(1-P_{in}^1)$

métodos aproximados [9]. De hecho, si nos centramos en el caso simple de sistemas multiservidor con una única órbita de reintentos, se puede destacar la ausencia de soluciones cerradas cuando $C > 2$ [3].

Obviamente, para resolver el sistema en estudio será necesario hacer uso de modelos aproximados y métodos numéricos de resolución. Aunque existen otras aproximaciones, para la comparación con VE se han escogido los tres métodos más conocidos aparecidos en la literatura hasta la fecha que son capaces de resolver el problema en cuestión. Estos métodos se explican en las siguientes subsecciones.

A. Truncación doble (DT)

El método más simple y sencillo para resolver el modelo propuesto consiste en la truncación de las dimensiones infinitas del espacio de estados [10]. En nuestro caso, debe aplicarse a las órbitas de remarcados y de reintentos automáticos, truncándolas a partir de los niveles Q_n y Q_h respectivamente y obteniéndose el siguiente espacio de estados:

$$\mathcal{S} := \{(k,m,o) : k \le C; m \le Q_n; o \le Q_h\}. \qquad (2)$$

Obviamente, incrementando los valores de Q_n y/o Q_h se aumenta el espacio de estados considerado y la precisión de la solución se espera que se incremente a costa de un incremento del coste computacional.

Las probabilidades de estado en régimen permanente pueden obtenerse resolviendo $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$ junto con la condición de normalización. Puesto que $\mathbf{Q}$ es una matriz finita este sistema puede resolverse con cualquiera de los métodos definidos en el álgebra lineal [11].

B. FM doble (DFM)

Al igual que DT, DFM pertenece a la familia de los métodos truncados finitos [3]. Estos métodos consisten en reemplazar el espacio de estados original infinito por uno finito. Sin embargo, DFM es más sofisticado que DT puesto que añade en algún sentido el efecto de los estados truncados.

En [12] desarrollamos FM, una generalización del método aproximado propuesto en [13]. Aunque inicialmente fue desarrollado para un escenario con una única órbita de reintentos, FM se aplicó con éxito a un sistema como el que se considera ahora en [14]. En este caso FM se ha aplicado tanto a las órbita de remarcados como a la de reintentos automáticos —convirtiéndose en DFM—, reduciendo el espacio de estados a un conjunto finito agregando todos los estados más allá de una ocupación de las órbitas dada, produciendo el mismo espacio de estados aproximado que DT:

$$\mathcal{S} := \{(k,m,o) : k \le C; m \le Q_n; o \le Q_h\}. \qquad (3)$$

donde Q_n (Q_h) define la ocupación a partir de la cual se agregan los estados de la órbita de remarcados (reintentos automáticos). En este caso los estados de la forma $(\cdot, Q_n, \cdot)$ representan la situación en la cual hay por lo menos Q_n usuarios en la órbita de remarcados. Asimismo los estados $(\cdot,\cdot,Q_h)$ representan la situación en la que hay Q_h o más usuarios en la órbita de reintentos automáticos. Debido a esa agregación se introducen dos nuevos parámetros. El parámetro M_n denota el número medio de usuarios en la órbita de remarcados condicionado a los estados donde hay por lo menos Q_n usuarios en la órbita, es decir, $M_n = E(m|m \ge Q_n)$. La probabilidad de que después de un remarcado exitoso el número de usuarios en la órbita de remarcados no esté por debajo de Q_n se representa mediante p_n. Para la órbita de reintentos automáticos se definen los parámetros M_h y p_h de forma análoga.

Las ecuaciones de balance globales, la ecuación de normalización y las ecuaciones para los parámetros M_n, p_n, M_h, p_h forman un sistema de ecuaciones que puede ser resuelto, por ejemplos mediante el proceso iterativo que se muestra en [14].

C. Truncación y generalización (TNR)

Mientras que los dos métodos aproximados anteriores consideran un método truncado finito para cada órbita de reintentos, este método considera el uso de un método truncado generalizado en una de las órbitas. Obviamente, no se puede usar un método generalizado en las dos órbitas de reintentos puesto que el modelo resultante no sería resoluble. Por este motivo, hemos aplicado un método truncado generalizado para la órbita de reintentos automáticos y una truncación para la órbita de remarcados. El método truncado generalizado escogido ha sido el propuesto por Neuts y Rao en [15]. Este método se basa en la homogeneización del

modelo más allá de un determinado nivel Q_h, lo cual supone restringir la tasa máxima de reintentos automáticos, es decir,

$$\mu_{ret}(o) = \begin{cases} o\mu_{ret} & \text{if } o < Q_h \\ Q_h\mu_{ret} & \text{if } o \geq Q_h \end{cases}$$

Por lo tanto, el espacio de estados resultante está definido por

$$\mathcal{S} := \{(k, m, o) : k \leq C; m \leq Q_n; o \in \mathbb{Z}_+\} \quad (4)$$

Con estas dos aproximaciones hay que resolver un sistema cuyo espacio de estados presenta dos dimensiones finitas y otra infinita, siendo la dimensión infinita homogénea a partir del nivel Q_h. Así, podemos resolver el sistema resultante y obtener las probabilidades de estado en régimen permanente haciendo uso de las soluciones geométrico matriciales para modelos estocásticos propuestas por Neuts en [8].

IV. VALUE EXTRAPOLATION (VE)

Todos los métodos descritos en la sección anterior se basan en el cálculo de las probabilidades de estado a partir de la resolución de las ecuaciones de balance. Sin embargo, muy recientemente Leino et al. [5] han propuesto una aproximación alternativa para la evaluación de procesos de Markov con un espacio de estados infinito. Esta aproximación, denominada *Value Extrapolation* —VE—, no se basa en la probabilidad de estar en un determinado estado, sino en una nueva métrica denominada *relative state values*, que aparece cuando consideramos el sistema como un MDP. Formalmente, un MDP puede definirse como la tupla $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}\}$, donde $\mathcal{S}$ es un conjunto de estados, $\mathcal{A}$ es un conjunto de acciones, $\mathcal{P}$ es una función de transición entre estados y $\mathcal{R}$ es una función de recompensa. El estado del sistema puede controlarse eligiendo acciones a de $\mathcal{A}$, lo cual tiene influencia en las transiciones de estado. La función de transición $\mathcal{P} : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}_+$ especifica la tasa de transición a otros estados cuando se toma una determinada acción en un estado dado. La primera característica de la técnica VE es la definición de una función de recompensa como función del estado del sistema, es decir, $r(s)$. A partir de la definición de la función de recompensa para cada estado, también se tiene una recompensa del proceso entero, r, que será el parámetro de mérito que se desea calcular.

Una vez definido un MDP y las funciones de recompensa, se pueden definir los *relative state values*. Es obvio que después de ejecutar una acción en el estado s el sistema obtendrá una recompensa por haber realizado esa acción $r(s)$, pero, conforme el número de transiciones aumenta, la recompensa media converge a r. El *relative state value* $v(s)$ expresa la diferencia entre la recompensa total cuando el sistema empieza en el estado s y la obtenida cuando la recompensa en todos los estados es r. Si denotamos mediante t_n los instantes de tiempo en los cuales hay un cambio de estado en el sistema, entonces

$$v(s) = E\left[\sum_{n=0}^{\infty} (r(S(t_n)) - r)\Big| S(t_0) = s\right]. \quad (5)$$

Las ecuaciones que relacionan recompensas, *relative state values* y tasas de transición son las ecuaciones de Howard, definidas por:

$$r(s) - r + \sum_{s'} q_{ss'}(v(s') - v(s)) = 0 \quad \forall s. \quad (6)$$

Habrán tantas ecuaciones de Howard como número de estados, $|\mathcal{S}|$. El número de incógnitas será el número de *relative state values* —$|\mathcal{S}|$— más la recompensa esperada r, es decir, $|\mathcal{S}| + 1$ incógnitas. Puesto que únicamente aparecen diferencias de *relative state values* en las ecuaciones de Howard, podemos definir $v(\mathbf{0}) = 0$, de modo que se obtendrá un sistema lineal de ecuaciones resoluble con el mismo número de ecuaciones que de incógnitas.

Sin embargo, se necesita un número finito de ecuaciones de Howard para resolver el sistema y, por lo tanto, hay que truncar el espacio de estados a $\hat{\mathcal{S}}$. Mientras que la truncación tradicional consiste en hacer $q_{ss'} = 0 \quad \forall s' \notin \hat{\mathcal{S}}$, VE realiza una truncación más eficiente. Básicamente, VE considera los *relative state values* que no pertenecen a $\hat{\mathcal{S}}$ y que aparecen en las ecuaciones de Howard como una extrapolación de algunos *relative state values* dentro de $\hat{\mathcal{S}}$. El objetivo de VE es encontrar una función $f(s)$ que interpole algunos puntos $(s, v(s))$ para $s \in \hat{\mathcal{S}}$ de modo que aproxime también $(s, v(s))$ para $s \notin \hat{\mathcal{S}}$. Es importante escoger una función de ajuste $f(s)$ que haga que las ecuaciones de Howard sigan siendo un sistema lineal de ecuaciones. Las funciones de ajuste más comunes que consiguen este propósito son los polinomios. Podemos usar todos los pares $(s, v(s))$ del espacio de estados en el proceso de ajuste —ajuste global— o únicamente un subconjunto ($\mathcal{S}_f$) de ellos —ajuste local. La elección de $\mathcal{S}_f$ dependerá asimismo del *relative state value* que queramos extrapolar. Nótese también que la función $f(s)$ y el conjunto $\mathcal{S}_f$ deben escogerse de forma no ambigua, es decir, en el caso de escoger un polinomio como función de ajuste, el número de pares $(s, v(s))$ diferentes en $\mathcal{S}_f$ debe ser igual o superior al número de coeficientes del polinomio. Nótese que si lo *relative state values* de fuera de $\hat{\mathcal{S}}$ fueran correctamente extrapolados, los resultados obtenidos al resolver el modelo truncado serían exactos.

A. Ecuaciones de Howard del sistema

Para obtener las ecuaciones de Howard del sistema estudiado vamos a clasificarlas en cuatro categorías atendiendo al número de servidores ocupados (k). Así tenemos:

1) $\mathbf{k < L}$: Estados en que se aceptan tanto sesiones nuevas como *handovers*. Las tasas de transición que salen de estos estados se representan en la Fig. 2. De este modo, las ecuaciones de Howard relacionadas con estos estados vienen dadas por:

$$\begin{aligned} r(k,m,o) - r + \lambda[v(k+1,m,o) - v(k,m,o)] + \\ + k\mu[v(k-1,m,o) - v(k,m,o)] + \\ + m\mu_{red}[v(k+1,m-1,o) - v(k,m,o)] + \\ + o\mu_{ret}[v(k+1,m,o-1) - v(k,m,o)] = 0. \end{aligned} \quad (7)$$

2) $\mathbf{k = L}$: Estados en que los *handovers* son aceptados siempre, mientras que las sesiones nuevas se aceptan con probabilidad $f = t - L$, donde t es el parámetro que caracteriza la política de control de admisión FGC. La Figura 3 muestra las tasa de transición desde estos estados; de donde se obtienen las siguientes ecuaciones de Howard:

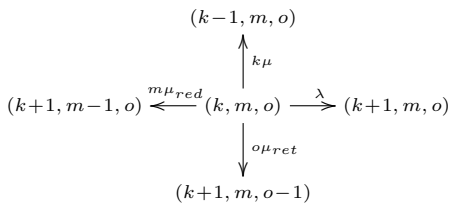


Fig. 2. Tasas de transición cuando $k < L$.

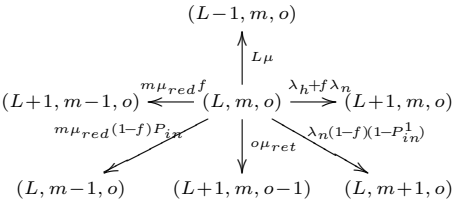


Fig. 3. Tasas de transición cuando $k = L$.

$$\begin{aligned} r(L,m,o) - r + \\ +(\lambda_h + \lambda_n f)[v(L+1,m,o) - v(L,m,o)] + \\ +L\mu[v(L-1,m,o) - v(L,m,o)] + \\ +m\mu_{red} f[v(L+1,m-1,o) - v(L,m,o)] + \\ +m\mu_{red}(1-f)P_{in}[v(L,m-1,o) - v(L,m,o)] + \\ +o\mu_{ret}[v(L+1,m,o-1) - v(L,m,o)] + \\ +\lambda_n(1-f)(1-P^1_{in})[v(L,m+1,o) - v(L,m,o)] = 0. \end{aligned} \quad (8)$$

3) $\mathbf{L < k < C}$: Estados en que únicamente se aceptan *handovers*, tal y como se observa en la Fig. 4. Las ecuaciones de Howard para estos estados vienen dadas por:

$$\begin{aligned} r(k,m,o) - r + \lambda_h[v(k+1,m,o) - v(k,m,o)] + \\ +k\mu[v(k-1,m,o) - v(k,m,o)] + \\ +m\mu_{red}P_{in}[v(k,m-1,o) - v(k,m,o)] + \\ +o\mu_{ret}[v(k+1,m,o-1) - v(k,m,o)] + \\ +\lambda_n(1-P^1_{in})[v(k,m+1,o) - v(k,m,o)] = 0. \end{aligned} \quad (9)$$

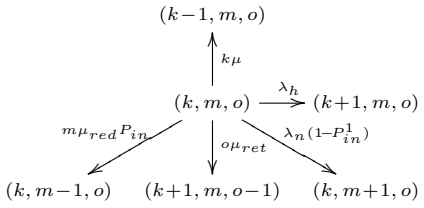


Fig. 4. Tasas de transición para $L < k < C$.

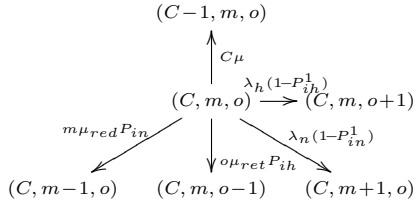


Fig. 5. Tasas de transición para $k = C$.

4) $\mathbf{k = C}$: Estados en que tanto sesiones nuevas como *handovers* son bloqueados, obteniéndose las transiciones que se observan en la Fig. 5 y las ecuaciones de Howard:

$$\begin{aligned} r(C,m,o) - r + \lambda_h(1-P^1_{ih})[v(C,m,o+1) - v(C,m,o)] + \\ +C\mu[v(C-1,m,o) - v(C,m,o)] + \\ +m\mu_{red}P_{in}[v(C,m-1,o) - v(C,m,o)] + \\ +o\mu_{ret}P_{ih}[v(C,m,o-1) - v(C,m,o)] + \\ +\lambda_n(1-P^1_{in})[v(C,m+1,o) - v(C,m,o)] = 0. \end{aligned} \quad (10)$$

Tabla II
DEFINICIÓN DE LA FUNCIÓN DE RECOMPENSA.

Parámetro	Valor
P_b^h	$r(k,m,o) = 1$ para $k = C$, $\forall m$, $\forall o$
	$r(k,m,o) = 0$ en otro caso
P_b^n	$r(k,m,o) = 1 - f$ para $k = L$, $\forall m$, $\forall o$
	$r(k,m,o) = 1$ para $k \geq L$, $\forall m$, $\forall o$
	$r(k,m) = 0$ en otro caso
N_{ret}	$r(k,m,o) = o$ $\forall k$, $\forall m$, $\forall o$
N_{red}	$r(k,m,o) = m$ $\forall k$, $\forall m$, $\forall o$

B. Función de recompensa

Para calcular los parámetros de prestaciones es necesario hacer uso de una función de recompensa. Así, se deben configurar los parámetros de entrada $r(s)$ de las ecuaciones de Howard de forma adecuada para que la tasa de recompensa del proceso completo, r, sea igual al parámetro de prestaciones que se quiere calcular. Dicho de otro modo, r representará el parámetro de prestaciones que queremos calcular si hacemos que $r(s)$ tome el valor que tendría dicho parámetro de prestaciones cuando el sistema se encuentra en el estado s. La Tabla II ofrece varios ejemplos de configuración del parámetro $r(s)$ para obtener los parámetros de prestaciones que se han utilizado en este estudio.

C. Ajuste polinómico y solución

Nótese que el sistema estudiado presenta un número de estados infinito puesto que tanto m como o pueden tomar cualquier valor en $\mathbb{Z}_+$, por lo que es necesario hacer uso de algún tipo de truncación. Se ha empleado una truncación similar a la de los métodos DT y DFM, obteniéndose un espacio de estados truncado definido por:

$$\hat{\mathcal{S}} := \{s = (k,m,o) : k \leq C; m \leq Q_n; o \leq Q_h\}.$$

Por lo tanto, para el sistema estudiado, se ha truncado el espacio de estados para valores de ocupación de la órbita de remarcados (reintentos automáticos) por encima de un valor Q_n (Q_h). Sin embargo, en las ecuaciones de Howard del espacio de estados truncado aparecen los *relative state values* de algunos estados que no pertenecen a la truncación. Se trata en concreto de $v(C,m,Q_h+1)$ $\forall m$ y $v(k,Q_n+1,o)$ para $k \geq L$ y $\forall o$. Será necesario extrapolar estos dos conjuntos de estados para obtener un sistema cerrado de ecuaciones. Para ello se ha utilizado un polinomio de grado $(n-1)$ que interpola los n puntos en $\{(j,v_j)|v_j = v(C,m,j), \forall m, Q_h - n < j \leq Q_h\}$ para extrapolar $v(C,m,Q_h+1)$. Para extrapolar $v(k,Q_n+1,o)$ para $k \geq L$ se interpolan los p puntos en $\{(i,v_i)|v_i = v(k,i,o), k \geq L, Q_n - p < i \leq Q_n, \forall o\}$. Nótese

que incluyendo los valores extrapolados no aumenta el coste computacional, ni el número de ecuaciones de Howard, que sigue siendo $|\hat{\mathcal{S}}| = (C+1) \times (Q_n+1) \times (Q_h+1)$.

Tras algunas operaciones algebraicas y haciendo uso de las bases de Lagrange para reducir la complejidad del proceso, se obtiene una sencilla expresión cerrada para calcular los valores extrapolados de ambos conjuntos de estados:

$$v(C,m,Q_h+1)^{(n)} = \sum_{j=0}^{n-1} (-1)^j \binom{n}{j+1} v(C,m,Q_h-j), \forall m,$$

y

$$v(k,Q_n+1,o)^{(g)} = \sum_{i=0}^{g-1} (-1)^i \binom{g}{i+1} v(k,Q_n-i,o), k \geq L, \forall o.$$

V. RESULTADOS

En esta sección se presentan una serie de ejemplos numéricos con el propósito de ilustrar las prestaciones y versatilidad de este modelo y de la metodología de análisis. El análisis numérico se ha enfocado también a la comparación entre la metodología propuesta y otras aproximaciones, no sólo en términos de precisión, sino también de coste computacional.

Para los experimentos numéricos se ha empleado una configuración básica del sistema sobre la que se van modificando diferentes parámetros. Así, salvo que se indique lo contrario, el valor de los diferentes parámetros del sistema será: $C = 10$, $t = 9$, $\mu = 1$, $P_{ih}^1 = P_{in}^1 = 0$, $P_{ih} = P_{in} = 0.2$, y $\mu_{red} = \mu_{ret} = 1$. Los valores de λ_n y λ_h se han modificado a través de la carga del sistema, $\rho = \lambda / C\mu$, siendo $\lambda = \lambda_n + \lambda_h$ y tomando $\lambda_h = 2\lambda_n$ en todos los casos. Se debe tener en cuenta que, dada la introducción del fenómeno de la impaciencia mediante los parámetros P_{in}^1, P_{in}, P_{ih}^1 y P_{ih}, se podrán considerar valores de $\rho > 1$.

A. Prestaciones de VE

El objetivo de esta sección es estudiar las prestaciones de diferentes polinomios de extrapolación en un amplio rango de escenarios. Obviamente, tal y como se ha comentado en la Sección III, para el sistema estudiado no somos capaces de calcular el valor exacto de los parámetros de prestaciones más comunes. Por esta razón, el primer estado es asumir que el valor exacto se puede obtener eligiendo un punto de truncación suficientemente elevado. Más concretamente, se han ejecutado todos los métodos presentados en la Sección III y VE, aumentando el punto de truncación, hasta que el valor de todos los parámetros de prestaciones estudiados se han estabilizado hasta el octavo decimal.

En el sistema estudiado, aparecen dos niveles de truncación, Q_n y Q_h, que deben ser especificados. El propósito de este estudio es determinar el par (Q_n, Q_h) que hace que la cardinalidad del problema ($(C+1)\times(Q_n+1)\times(Q_h+1)$) sea lo más pequeña posible a la vez que se cumple un determinado criterio de precisión. Para cumplir estos requisitos es necesario definir un proceso de búsqueda del par (Q_n, Q_h).

Con el fin de evitar una búsqueda exhaustiva para determinar (Q_n, Q_h) se ha utilizado un algoritmo similar al propuesto en [16]. Nuestro algoritmo aumenta (Q_n, Q_h) a lo largo de la diagonal hasta que el sistema cumple el requisito de precisión especificado. Posteriormente se disminuye el valor de cada uno de los parámetros de forma separada siguiendo las direcciones descendientes de los ejes de coordenadas. Finalmente se toma la mejor solución en términos de la cardinalidad del problema. La lógica tras este último movimiento para uno de los parámetros (Q_n o Q_h) se encuentra en que generalmente $Q_n \neq Q_h$, y este hecho no puede conseguirse sólo con el movimiento a lo largo de la diagonal. Así, la solución con este último movimiento descendente mejora, en términos de cardinalidad, el movimiento inicial a través de la diagonal.

En la Tabla III se muestra la complejidad mínima del problema necesaria para asegurar un error relativo menor que 10^{-4} en los parámetros P_b^n y P_b^h, para diferentes cargas del sistema (ρ), diferentes tasas de reintento ($\{\mu_{red}, \mu_{ret}\}$) y diferentes grados del polinomio de extrapolación.

Nótese que VEx denota el uso de un polinomio de extrapolación de grado x. Nótese asimismo que los números mostrados en cada celda representan el producto $(Q_n+1)\times(Q_h+1)$ que define la complejidad y que se denota como Ω. Aunque la cardinalidad del problema debería incluir el factor $(C+1)$, este se ha omitido ya que es común a todos los casos. De este modo, el mejor polinomio será aquel que presente el menor valor de Ω, dicho valor se muestra en negrita en la tabla. Además se denotan como "-" aquellos casos que no han podido resolverse por falta de memoria[1].

A partir de los resultados de la Tabla III se puede concluir que no hay un polinomio de extrapolación que resulte la mejor solución en los diferentes casos estudiados. Ni el polinomio de menor grado, ni el de mayor grado ofrecen los mejores resultados. Así, cuando la carga no es alta ($\rho = 0.4$), VE2 ofrece la menor complejidad, debido a que los polinomios VE3-VE6 ofrecen como resultado el menor Ω con el que son capaces de trabajar, es decir, para extrapolar con el polinomio VE4 es necesario usar, como mínimo $Q_n = Q_h = 4$ y por tanto, el Ω mínimo será $(4+1)\times(4+1) = 25$. Por otro lado, cuando las órbitas de reintento están más llenas, VE4 es una buena opción, puesto que ofrece valores bajos de Ω en todos los casos. Es por ello que, de ahora en adelante se utilizará un polinomio de grado 4 (VE4) para realizar la extrapolación, al que denominaremos simplemente VE.

B. Comparación entre métodos

1) Precisión: El objetivo de esta sección es comparar las prestaciones de VE con las de DT, DFM y TNR. En la Tabla IV se muestran los valores mínimos de Ω necesarios para obtener un error relativo menor que 10^{-4} en N_{red}. Los resultados para el resto de parámetros se han omitido puesto que N_{red} es generalmente el caso peor para todos los métodos comparados. Además los resultados obtenidos son cualitativamente equivalentes para todos los parámetros de prestaciones. Nótese que se muestra en negrita el mejor resultado, es decir, aquel que ofrece una complejidad mínima. Los resultados muestran que VE mejora claramente los resultados obtenidos por el resto de métodos puesto que requiere un valor de Ω mucho menor para conseguir la precisión deseada en todos los escenario estudiados. Además, y lo que puede ser aún más importante, existen escenarios en que VE es el único método capaz de obtener resultados.

[1]Los resultados han sido obtenidos usando Matlab sobre un Intel Core 2 Quad Q6600 con 4GB de memoria RAM.

Tabla III
Ω MÍNIMA PARA OBTENER UN ERROR RELATIVO MENOR QUE 10^{-4} EN P_b^n/P_b^h.

μ_{red}, μ_{ret}	ρ	VE1	VE2	VE3	VE4	VE5	VE6
{1,1}	0.4	25/30	**12**/**12**	16/16	25/25	36/36	49/49
	0.8	144/144	49/72	64/72	49/**35**	**36**/36	49/49
	1.2	484/506	342/342	240/**36**	**98**/120	121/132	99/120
{2,0.5}	0.4	20/25	**12**/**12**	16/16	25/25	36/36	49/49
	0.8	130/90	45/55	56/64	**36**/**30**	**36**/36	49/49
	1.2	-/-	432/336	280/170	**99**/**136**	126/144	135/168
{0.5,2}	0.4	20/25	**12**/**12**	16/16	25/25	36/36	49/49
	0.8	160/160	66/110	80/100	56/49	**36**/**42**	49/49
	1.2	-/-	-/-	400/-	154/189	**144**/**187**	162/198
{0.5,0.5}	0.4	25/30	**9**/**9**	16/16	25/25	36/36	49/49
	0.8	224/160	100/121	90/100	48/**35**	**36**/36	49/49
	1.2	-/-	-/-	-/-	**168**/280	195/**196**	441/378

Tabla IV
Ω MÍNIMO PARA OBTENER UN ERROR RELATIVO MENOR QUE 10^{-4} EN N_{red}.

	$\mu_{red}, \mu_{ret} = \{1,1\}$					$\mu_{red}, \mu_{ret} = \{2,0.5\}$					$\mu_{red}, \mu_{ret} = \{0.5,5\}$					$\mu_{red}, \mu_{ret} = \{0.5,0.5\}$				
ρ	0.4	0.6	0.8	1.0	1.2	0.4	0.6	0.8	1.0	1.2	0.4	0.6	0.8	1.0	1.2	0.4	0.6	0.8	1.0	1.2
DT	64	143	324	550	930	56	132	304	522	-	54	120	264	-	-	63	180	528	-	-
DFM	48	72	208	360	324	49	100	176	378	-	45	98	198	-	-	56	126	352	-	-
TNR	48	91	180	400	651	48	99	182	196	640	36	90	192	-	-	54	135	240	-	-
VE	**25**	**25**	**35**	**110**	**196**	**25**	**25**	**35**	**108**	**204**	**25**	**25**	**60**	**66**	**161**	**25**	**25**	**45**	**195**	**396**

2) Coste computacional: Aunque VE mejora claramente los resultados obtenidos por los otros métodos en términos de precisión, es interesante estudiar también el coste computacional asociado. Desde un punto de vista práctico, es más interesante considerar la precisión junto con el tiempo de cómputo. Así, la Fig. 6 muestra una representación conjunta de ambos parámetros. Los resultados deben interpretarse cuidadosamente, puesto que el coste computacional depende en gran medida del algoritmo utilizado para resolver el sistema de ecuaciones resultante. Más concretamente, para calcular la matriz $\mathbf{R}$ que aparece en TNR se ha utilizado el algoritmo de reducción logarítmica propuesto en [17, Sección 8.4], usando una precisión de 10^{-6} en el proceso iterativo. Además, para resolver los sistemas resultantes de los métodos DT, DFM y TNR se ha usado el algoritmo descrito en [11] y que aprovecha la estructura tridiagonal a bloques que presenta el generador infinitesimal. Desgraciadamente, el sistema de ecuaciones lineales que se obtiene en VE no presenta estructura tridiagonal a bloques y, por tanto, es necesario hacer uso de un algoritmo más general. En este caso se ha hecho uso de la factorización LU.

Se puede observar como el tiempo necesario para resolver el sistema con cualquiera de los métodos empleados no es muy alto desde un punto de vista humano. Por esa razón, los resultados temporales deben compararse cualitativamente, puesto que las unidades de tiempo pueden ser mucho mayores en el caso de resolver sistemas más complejos o cuando se debe resolver un sistema sencillo una gran cantidad de veces

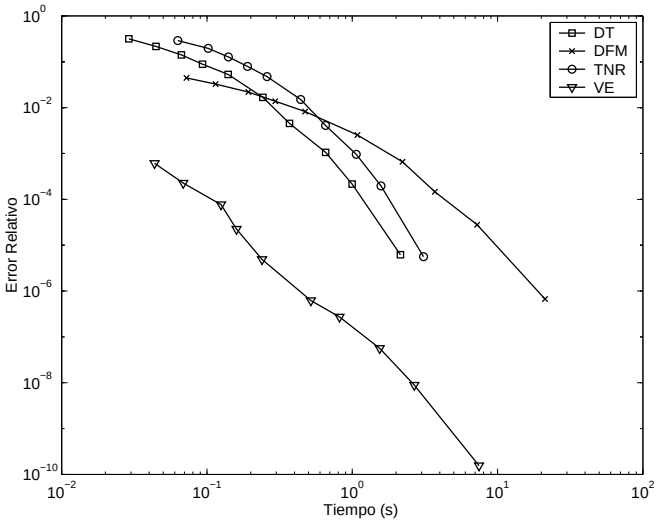


Fig. 6. Tiempo de cómputo para los diferentes métodos.

— por ejemplo, para balancear la tasa de *handovers* entrantes en la célula con la tasa de salida de *handovers*, tal y como se muestra en [18]—.

VI. CONCLUSIONES

En sistemas de comunicaciones móviles como las redes celulares, *Mobile IP* o los recientemente definidos IEEE 802.16e y IEEE 802.20, los operadores de red deben garantizar la movilidad sin interrupciones a sus clientes. En

estas redes, los reintentos ocurren debido al remarcado de los usuarios cuando los intentos de establecer una nueva sesión son bloqueados, y también debido a los reintentos automáticos cuando un *handover* falla. El modelo markoviano que describe esta red es un sistema de reintentos multiservidor que presenta heterogeneidad en el espacio de estados a lo largo de dos dimensiones infinitas. Hasta donde nosotros conocemos, todos los métodos estudiados en la literatura para resolver estos sistemas se basan en el cálculo de las probabilidades de estado en régimen permanente. En este artículo se propone un método alternativo basado en otra métrica: los *relative state values* y las ecuaciones de Howard que los relacionan.

Se ha comparado el método propuesto con las aproximaciones más importantes que han aparecido en la literatura hasta el momento. Los resultados muestran que el método propuesto generalmente mejora las aproximaciones anteriores no sólo en términos de precisión sino también en coste computacional. Además, se ha mostrado como en algunos escenarios este método es el único capaz de garantizar una determinada precisión. Por todo ello el método propuesto es muy recomendable para resolver este tipo de sistemas.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el Gobierno de España (30% PGE) y la Comisión Europea (70% FEDER) a través de los proyectos TSI2007-66869-C02-02 y TIN2008-06739-C04-02.

REFERENCIAS

[1] M. Mouly y M.B. Pautet, “The GSM system for mobile communications”. Publicado por los autores, 1992.

[2] E. Onur, H. Deliç, C. Ersoy y M.U. Çağlayan. “Measurement-based replanning of cell capacities in GSM networks”, *Computer Networks* Vol. 39 pp. 749–767, 2002.

[3] J.R. Artalejo y M. Pozo, “Numerical calculation of the stationary distribution of the main multiserver retrial queue”, *Annals of Operations Research*, Vol. 116, no. 1–4, pp. 41–56, 2002.

[4] M.J. Domenech-Benlloch, J.M. Gimenez-Guzman, V. Pla, J. Martinez-Bauset y V. Casares-Giner, “Generalized Truncated Methods for an Efficient Solution of Retrial Systems”, *Mathematical Problems in Engineering* Vol. 2008, Article ID 183089, 2008.

[5] J. Leino, A. Penttinen y J. Virtamo, “Flow level performance analysis of wireless data networks: A case study”, *Proceedings of IEEE ICC*, Vol. 3, pp. 961–966, 2006.

[6] F. Khan y D. Zeghlache, “Effect of Cell Residence Time Distribution on the Performance of Cellular Mobile Networks”, *Proceedings of IEEE VTC* pp. 949–953, 1997.

[7] R. Ramjee, R. Nagarajan y D. Towsley, “On optimal call admission control in cellular networks”, *Wireless Networks Journal (WINET)*, Vol. 3, no. 1, pp. 29–41, 1997.

[8] M. Neuts, *Matrix-geometric Solutions in Stochastic Models: An Algorithmic Approach*. The Johns Hopkins University Press 1981.

[9] G. Falin y J. Templeton, *Retrial Queues*. Chapman & Hall 1997.

[10] R.I. Wilkinson, “Theories for toll traffic engineering in the USA”, *The Bell System Technical Journal*, Vol. 35, no. 2, pp. 421–514, 1956.

[11] L.D. Servi, “Algorithmic solutions to two-dimensional birth-death processes with application to capacity planning”, *Telecommunication Systems*, Vol.21, no. 2–4, pp. 205–212, 2002.

[12] M.J. Domenech-Benlloch, J.M. Gimenez-Guzman, J. Martinez-Bauset y V. Casares-Giner, “Efficient and accurate methodology for solving multiserver retrial systems”. *IEE Electronic Letters*, Vol. 41, no. 17, pp. 967–969, 2005.

[13] M.A. Marsan, G.D. Carolis, E. Leonardi, R.L. Cigno y M. Meo, “Efficient estimation of call blocking probabilities in cellular mobile telephony networks with customer retrials”, *IEEE Journal on Selected Areas in Communications*, Vol. 19, no. 2, pp. 332–346, 2001.

[14] J.M. Gimenez-Guzman, M.J. Domenech-Benlloch, V. Pla, V. Casares-Giner y J. Martinez-Bauset, “Guaranteeing Seamless Mobility with User Redials and Automatic Handover Retrials”, *Journal of Universal Computer Science*, Vol. 14, no. 10, pp. 1597–1624, 2008.

[15] M. Neuts y B. Rao, “Numerical investigation of a multiserver retrial model”, *Queueing systems*, Vol. 7, pp. 169–190, 1990.

[16] J.R. Artalejo y V. Pla, “On the impact of customer balking, impatience and retrials in telecommunication systems”. *Computers & Mathematics with Applications*, Vol. 57, no. 2, pp. 217–229, 2009.

[17] G. Latouche y V. Ramaswami, *Introduction to Matrix Analytic Methods in Stochastic Modeling*. ASA-SIAM 1999.

[18] M.A. Marsan, G.D. Carolis, E. Leonardi, R.L. Cigno and M. Meo, “How many cells should be considered to accurately predict the performance of cellular networks?”, *Proceedings European Wireless* 1999.

Estudio a nivel de aplicación del consumo energético de 802.11 en teléfonos móviles

Estrella M García-Lozano, Celeste Campo, Carlos García-Rubio, Alberto Cortés
Departamento de Ingeniería Telemática,
Universidad Carlos III de Madrid
Avda de la Universidad, 30, 28911 Leganés (Madrid)
{emglozan, celeste, cgr, alcortes}@it.uc3m.es

***Resumen*—Hoy en día, los dispositivos móviles tales como teléfonos o PDAs nos acompañan siempre y nos son de gran utilidad. Si queremos maximizar el tiempo en el que están activos, es decir, retrasar lo más posible el agotamiento de la batería, debemos vigilar y reducir el consumo energético de una de las principales razones de gasto: la interfaz WLAN. Hemos medido el consumo de las comunicaciones UDP *unicast* en una situación muy habitual: un teléfono asociado a una red 802.11 estructurada. De esta forma, analizando y modelando los resultados, podremos entender cómo se produce el consumo y seleccionar comportamientos optimizados para cualquier aplicación que queramos construir sobre dicho tipo de sistema.**

***Palabras Clave*—802.11, consumo energético, modelado, UDP, guías de diseño**

I. Introducción

Actualmente, nos encontramos rodeados de pequeños dispositivos móviles, tales como los teléfonos móviles y las PDAs, que se enriquecen constantemente con nuevas funcionalidades. Debido a su moviliad, deben depender de baterías y la experiencia del usuario dependerá de la frecuencia con que debe recargarlas. Puesto que la evolución en la capacidad de las baterías no avanza al mismo ritmo que las necesidades impuestas por el resto de tecnologías presentes en un dispositivo móvil [1], el ahorro de energía se hace crucial. Sabiendo que la interfaz WLAN es una de las partes del dispositivo móvil que más consumen, tal y como ya se vio en [2], pensamos que se debe poner esfuerzo en aplicar técnicas de ahorro energético en ella.

Por lo tanto, el primer paso es conseguir una visión completa y precisa de los patrones de gasto de la interfaz WLAN. Algunos dispositivos, como los ordenadores portátiles, ya han sido estudiados, por ejemplo en [3] y [4]. Nuestro trabajo se centra en un aparato tan popular como el teléfono móvil. Hemos observado su consumo en transmisión, en recepción y en estado *idle* (conectado pero sin intercambio de datos), adoptando diferentes comportamientos.

Una vez que sabemos lo que es mejor y lo que es peor en términos de gasto energético, podemos evaluar y proponer medidas para la eficiencia energética en protocolos existentes y futuros. Como un ejemplo de esto, revisaremos las recomendaciones de la RFC 5405 del IETF *Unicast UDP Usage Guidelines for Application Designers* [5]. Esta RFC, que data de Noviembre de 2008, es el fruto de un grupo de trabajo muy activo y consiste en una compilación de guías acerca de varios temas relacionados con el uso de UDP: control de congestión, tamaño del mensaje, confiabilidad, *checksum*, paso por *middleboxes*, programación e ICMP. Los autores están especialmente interesados en evitar la congestión en la red, y muchas de las guías de uso están muy orientadas en este sentido. Un análisis del consumo de potencia puede añadir eficiencia en cuanto a tiempo de vida de los dispositivos en algunos puntos. En este artículo se tratarán el tamaño de mensaje y el paso por *middleboxes* (como puede ser un NAT, un caso muy común actualmente).

Así, en primer lugar presentaremos el trabajo relacionado con el tema de estudio y dónde podemos añadir algo de valor. Después, explicaremos la primera parte de nuestra investigación, que es un estudio del consumo en teléfonos móviles. En esta sección, primero hablaremos sobre algunos puntos del estándar 802.11 relacionados con el consumo energético, y terminaremos esta parte con una explicación de los resultados obtenidos junto con algunas conclusiones sobre ellos. En la siguiente sección haremos una revisión de los dos puntos de la RFC 5405 mencionados antes: el tamaño de paquete y el paso por *middleboxes*, ya que pueden ser mejorados con algunas recomendaciones sobre eficiencia energética. Por último, presentaremos nuestras conclusiones y propuestas para futuros trabajos a partir de esta investigación.

II. Trabajo relacionado

Fue en [3] donde se sentaron las bases de la investigación sobre el consumo energético. Los autores tomaron medidas del consumo en ordenadores portátiles para crear un modelo lineal simple que fuera capaz de representar los resultados obtenidos. En este modelo, la energía consumida por paquete depende del tamaño del mismo y más un coste fijo, ambos dependientes del tipo de comunicación: transmisión, recepción, recepción promiscua o descarte; punto a punto o difusión; nodo no destinatario en el área de influencia de la fuente, del destino, de ambos o de ninguno. Su formulación es ésta:

$$Energy = m \times size + b \quad (1)$$

También demostraron que, para los posibles estados de la tarjeta de red, esto es: apagada, dormida (*sleep*), ociosa (*idle*), recibiendo y transmitiendo, el consumo es cada vez mayor según este mismo orden.

Un modelo algo más complejo fue desarrollado en [6] utilizando los resultados provistos por el anterior artículo. En [7] podemos ver un trabajo similar con un modelo lineal paralelo, esta vez desde el punto de vista de la capacidad de la batería.

Sin embargo, en [8] los autores muestran que para niveles de potencia de transmisión bajos el modelo ya descrito deja de ser aplicable, y proponen otro algo más complejo pero aún lineal. En [9] encontramos el mismo cambio, puesto que para

niveles de potencia de transmisión altos se obtienen resultados contrarios a los conseguidos con niveles bajos. También relacionan el consumo con el tamaño de paquete, haciendo hincapié en que tamaños de paquete pequeños implican un consumo por byte más alto, dados los costes fijos.

Toda la literatura mencionada utilizó ordenadores portátiles para sus experimentos, principalmente en entornos *ad hoc*, midiendo el consumo directamente de la tarjeta de red. Hay menos trabajos que tomen medidas de dispositivos de pequeño tamaño, ni que midan el consumo global del dispositivo cuando se está haciendo uso de la WLAN y no sólo de la tarjeta de red aislada.

En [10] los autores usan dos PDAs y sus resultados son muy similares a los de [3]. También con PDAs pero en este caso haciendo uso del API del SO, en [11] se muestran datos no sólo sobre el modo *ad hoc* sino también en modo estructurado, y sobre *multicast* además de *unicast*. En [4] el consumo de un ordenador portátil se mide de forma global, y se hace un estudio interesante de su modelo de energía desde el punto de vista de una aplicación.

En cuanto a la RFC de estudio, está en la base de un desarrollo actual como es XPP (Extensible Peer Protocol [12]), el cual va un poco más allá de UDP. Otros protocolos en desarrollo que consideran el uso de UDP y prestan atención a las guías proporcionadas son TURN (Traversal using relays around NATs [13]) e IPFIX (IP Flow Information Export [14]). El grupo de trabajo MIPSHOP también lo tiene en cuenta para el diseño de un MSTP (Mobility Services Transport Protocol [15]) y permite el uso de UDP como tal. Las guías de uso de UDP en *unicast*, en este caso especial de redes móviles, no están completas sin la adición de recomendaciones acerca de la eficiencia energética.

Como podemos ver, todavía falta en la literatura más información acerca de dispositivos pequeños y por tanto de recursos limitados como las PDAs y los teléfonos móviles, siendo éstos los más propensos a formar parte de redes inalámbricas en cualquier momento. También encontramos que la mayoría de las soluciones para el ahorro de energía se sitúan en el nivel de enlace o en un nivel intermedio entre éste y el de red. Sin embargo, hay poca investigación en curso al nivel de aplicación para este tipo de dispositivos, sabiendo que un comportamiento poco apropiado de este nivel puede hacer que la batería se acabe rápidamente. Es en este sentido, también, donde las guías sobre UDP están incompletas.

III. Modelo energético de IEEE 802.11 para teléfonos móviles

Queremos caracterizar el consumo energético a nivel de aplicación de un teléfono móvil cuando se utiliza la interfaz de WLAN. Primero necesitamos saber algunos puntos básicos acerca del estándar, que nos ayudarán a definir los objetivos y el modelo para los experimentos. A la luz de los resultados podremos sacar algunas conclusiones sobre los patrones de consumo encontrados.

A. Consumo energético en 802.11

En [3] se da una descripción completa de los patrones de gasto en 802.11:

Tarjeta apagada.: Cuando la tarjeta de red está apagada, no hay ningún gasto energético debido a ella, pero al mismo tiempo es imposible recibir o enviar paquetes.

Modo sleep.: En este estado, la tarjeta de red no está trabajando pero está lista para pasar al estado activo, de modo que pueda enviar o recibir un paquete. Este estado consume muy poco, cerca del consumo estando apagada.

Estado idle.: Cuando la tarjeta está en este estado, está escuchando en la red de modo que pueda recibir un paquete entrante, aunque en ese momento no está recibiendo ni enviando nada. Puesto que es un estado activo, el consumo sube en comparación con el modo *sleep*.

Descarte.: Puede ocurrir que la tarjeta esté en estado *idle* y aparezca en la red un paquete destinado a otro. Aun así el paquete es escuchado y luego descartado. Escuchar este paquete implica un consumo algo mayor (aunque muy poco) con repecto a estar simplemente en estado *idle*.

Transmisión/Recepción.: Cuando un nodo de la red envía un paquete a otro, primero escucha la red y, si no hay ninguna transmisión en curso, envía un RTS (Request To Send) y espera una respuesta. Si el destino recibe el RTS y está listo, envía un CTS (Clear To Send). Entonces, el emisor puede transmitir el paquete y, después, el receptor deberá responder con un ACK (Acknowledgement). Si el tamaño del paquete está por debajo de un umbral (configurable) para el RTS, el intercambio de RTS/CTS puede consumir más recursos en la red que el mismo envío, por lo que este mecanismo para evitar colisiones puede ser suprimido. También puede ser totalmente desactivada. Tanto si existe RTS/CTS como si no, puesto que enviar precisa más potencia que recibir, la recepción será algo más costosa que el estado *idle* (y que el descarte debido al envío del ACK), pero sensiblemente menos que la transmisión.

El estándar también implementa su propia técnica de ahorro de potencia (PSM, *Power Saving Mode*). Se basa en el hecho de que, cuando el dispositivo está escuchando la red para posibles mensajes entrantes (estado *idle*) o descartando paquetes, consume más que cuando está durmiendo, incrementando así notablemente el consumo a largo plazo. Cuando todos los dispositivos están sincronizados con una estación base, o entre ellos en una red *ad hoc*, los mensajes que deben ser reenviados a su destino pueden ser almacenados temporalmente mientras que todos los nodos están durmiendo, y enviados después por turnos cuando despierten y entren en estado *idle*. El intervalo para dormir se llama intervalo de *beacon*, y los nodos que tienen paquetes en espera para ellos se anuncian en cada turno justo después del *beacon*, en un paquete especial llamado TIM (Traffic Indication Map), de modo que los nodos que no los tienen pueden volver a pasar a estado *sleep*.

Esta técnica, aunque efectiva, puede ser perjudicial para algunas aplicaciones, como pueden ser las multimedia, ya que podría interferir en la cadencia de envío deseada. Los dispostivos móviles normalmente ofrecen la posibilidad de activar o desactivar el modo de ahorro de energía, de modo que el usuario pueda elegir dependiendo de sus preferencias en ese momento.

Algunos autores han propuesto mejoras sobre este mecanismo según diferentes estrategias, como en [16], [17] y [18]. En muchos casos, la optimización en un nivel superior podría complementarlo y mejorar drásticamente el ahorro de energía.

B. Objetivos

Hay muchos aspectos que pueden afectar al consumo de potencia en un dispositivo WLAN. Gran cantidad de ellos provienen de la propia operación del dispositivo, como pueden ser técnicas de ahorro de energía de la tarjeta de red, otros vienen de la configuración de otros dispositivos en la red, y otros vienen del comportamiento de las aplicaciones que utilizan la interfaz WLAN. Queremos centrarnos en los primeros y en los últimos, más concretamente en los siguientes puntos:

- Consumo de potencia medio para envío, recepción y estado *idle*.
- Dependencia de la cadencia de envío/recepción, el tamaño de datos en el paquete y la tasa de datos.

Tenemos siempre en cuenta que la tarjeta de red no es la única responsable del consumo de batería, sino que la CPU y otras partes del dispositivo pueden tomar una parte significativa en ello (como se explicó en [19]). Por tanto, queremos medir no sólo el consumo de la tarjeta, sino el gasto de energía del dispositivo completo durante la comunicación WLAN. Es por esto que decidimos tomar las medidas de consumo de forma global en el dispositivo, con la contrapartida de que no podremos distinguir claramente la potencia consumida por la interfaz radio y por el resto de factores. Sin embargo, esto no tiene por qué ser un problema, puesto que ya hay estudios de dicha intefaz de forma aislada y en profundidad, por ejemplo en [3], en los que nos podemos apoyar llegados a ese punto.

La ventaja de tratar con el consumo total es que podemos buscar la eficiencia también a nivel de aplicación. Esto es lo que pretendemos cuando releemos la RFC 5405 desde un punto de vista centrado en el consumo. En ella, se observan otros aspectos, principalmente el control de congestión, para crear unas guías para desarrolladores de aplicaciones de red sobre UDP como protocolo de transporte, de modo que se haga un uso eficiente de la red. Añadiremos nuestras conclusiones sobre qué es mejor para reducir el gasto energético a algunas de las guías dadas, permitiendo a los desarrolladores mejorar energéticamente sus aplicaciones.

En la siguiente sección explicaremos las limitaciones que tuvimos que afrontar y el diseño de los experimentos para conseguir la información que necesitamos.

C. Diseño del experimento

Cuando se trabaja con sistemas empotrados, medir el consumo de componentes se convierte en una tarea difícil y muchas veces imposible, puesto que puede destruir el dispositivo. Por tanto, el modo de realizar mediciones se convierte a su vez en tema de investigación. Las medidas físicas pueden ser más precisas pero requieren la construcción de circuitos delicados ya que muchos teléfonos no funcionan sin tener la batería insertada. La otra opción es tomar medidas mediante herramientas *software*, y entonces dependerá de cómo de precisas son las medidas que el fabricante es capaz de proporcionar. Algunas plataformas proporcionan herramientas que normalmente acceden a funciones privadas para dar información más precisa.

Hay trabajo previo en el área de recoger información de consumo mediante aplicaciones, como se vio en [4] y [11], donde se utilizaba el API del sistema operativo para conseguir datos sobre el uso de la energía. Para Windows Mobile hay algunas herramientas disponibles, como acbPowerMeter [20] y acbTaskMan [21]. En el caso de Symbian, la plataforma que usamos en los experimentos, conseguir una visión más o menos precisa del gasto mediante APIs públicas del sistema operativo fue imposible, ya que sólo se puede consultar el nivel de batería en séptimas partes. Sin embargo, después del no tan útil *plugin* para su IDE Performance Investigator [22], en Diciembre de 2007 Nokia presentó Energy Profiler [23] (versión 1.1 a partir de Mayo de 2008 y 1.2 en Abril de 2009), una herramienta para Symbian que se ejecuta en el teléfono y proporciona información completa y abundante sobre el consumo energético y otros datos relacionados como la carga de la CPU, muy al estilo de acbTaskMan.

1) Metodología: El escenario para nuestros experimentos se representa en la Fig. 1. Usamos un punto de acceso que conectaba la red WLAN del teléfono con el segmento cableado en el que había un PC como interlocutor. Otro PC monitorizaba la actividad.

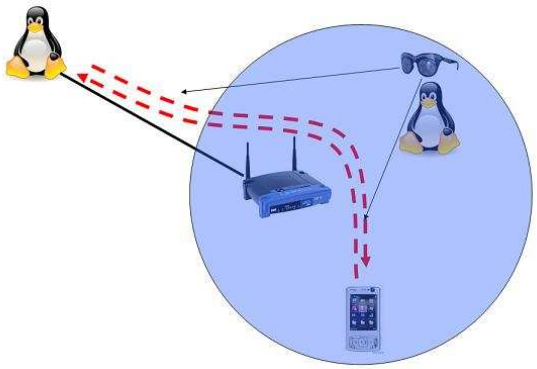


Fig. 1. Escenario del experimento.

Decidimos trabajar en modo infraestructura y generar tráfico inalámbrico enviando paquetes UDP porque nos permite comparar los resultados con la literatura previa. Más concretamente, diseñamos nuestro escenario de pruebas basándonos en los de [11], puesto que no podemos conseguir medidas muy precisas sin intervalos de muestreo por debajo de 1 segundo (límite impuesto por la herramienta de medición).

Una prueba consistía en esta secuencia:

1) Empieza la medición
2) El teléfono crea una asociación WLAN con el punto de acceso
3) Espera de cinco minutos
4) El teléfono comienza a enviar o recibir paquetes según una serie de parámetros: tamaño de paquete, longitud del intervalo entre paquetes y número de paquetes durante diez minutos
5) Cuando termina, otra espera de cinco minutos
6) Se destruye la asociación con el punto de acceso
7) Se para la medición

Como se señaló antes, las pruebas se definen por el tamaño de paquete (el máximo son 1472 bytes), la longitud del intervalo entre paquetes consecutivos (1729 ms como mucho), el número de paquetes (que debía hacer que el tiempo de comunicación del experimento durara 10 minutos) y el sentido de la transmisión (desde o hacia el teléfono). Estos valores

máximos son los correspondientes al envío de 500 KB de datos UDP en un tiempo total de 600 segundos, en tramas de 1500 bytes. Dichos parámetros eran manejados por una aplicación en el teléfono que se coordinaba con otra en el PC para establecer los términos de la comunicación y comenzarla. Después de cada prueba, calculábamos la potencia media del periodo de trabajo. En todas las gráficas de resultados se muestran estas medias y la desviación típica correspondiente a cada una.

Hay tres tipos de experimentos, como se puede ver en la Fig. 2:

- Tasa de datos fija.
- Tamaño de paquete fijo.
- Cadencia fija.

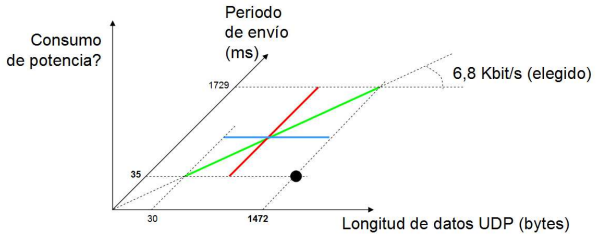


Fig. 2. Parámetros para los experimentos de transmisión/recepción.

Los tres se realizaron para transmisión y para recepción, más una prueba adicional en la que se creó una asociación con el punto de acceso pero se mantenía el estado *idle*.

2) Detalles: El dispositivo que utilizamos para la investigación es un Nokia N95, con versión de *firmware* 11.0.026, el cual dispone de tarjeta WLAN y un menú para su configuración. A través del menú de ajustes se puede establecer que trabaje en modo *ad hoc* o infraestructura, usando una dirección IP estática o no, y otros detalles del nivel IP. La potencia máxima de transmisión puede fijarse en 4, 10 o 100 mW, y el modo de ahorro de energía (PSM) puede ser habilitado y deshabilitado por el usuario. En nuestras pruebas, la potencia de transmisión se fijó en 4 mW, el PSM estaba desactivado y asignamos al teléfono una dirección IP estática.

El punto de acceso era un Linksys WRT54G versión 5.1 con el *firmware* de fábrica (versión 1.00.6). Algunos parámetros configurables interesantes son el intervalo de *beacon* (fijado en 100), el intervalo de DTIM (configurable de 1 a 255, a 1 para evitar que los clientes entren en modo PSM), el umbral de fragmentación (fijado al máximo, 2346 bytes, para que la MTU no tenga efecto visible), el umbral de RTS (fijado al máximo, 2347 bytes, para que nunca utilicemos la protección frente a colisiones, ya que no esperamos ninguna), la tasa de transmisión (fijada a "Auto", estando en modo 802.11b), y la potencia de transmisión (de 0 a 84 mW, fijada a 28 mW por defecto). Todo esto era accesible y configurable a través de su interfaz web.

El teléfono estaba a aproximadamente 2 metros del punto de acceso, por lo que podemos encontrar un incremento mínimo del consumo de potencia debido a potencias de transmisión altas. Los PCs eran puestos con tarjetas inalámbricas y cableadas que ejecutaban un Linux 2.6.18-6-686. La instalación para ambos era mínima para evitar demonios de red corriendo inadvertidamente.

tarjeta apagada	0.033 W
tarjeta en estado *idle*	1.071 W

Tabla I
POTENCIA MEDIA CONSUMIDA CUANDO LA TARJETA WLAN ESTÁ APAGADA Y EN ESTADO *idle*.

Por último, la versión de la herramienta de medición, Nokia Energy Profiler, es la 1.1. Hemos comprobado que la versión 1.0 provee valores de potencia consumida más bajos para el mismo experimento. Los intervalos de muestreo disponibles son 0.25, 1 y 5 segundos. Para conseguir medidas fiables, el fabricante desaconseja el intervalo de 0.25 s, por lo que elegimos el de 1 s a cambio. También se deben desactivar las capturas de pantalla automáticas y lanzar y cerrar la funcionalidad de estudio mientras que Energy Profiler está midiendo para conseguir las medidas más fiables.

D. Resultados

Primero, hay que averiguar el gasto debido a mantener el teléfono asociado al punto de acceso. En la Tabla I se puede ver la potencia media gastada por tener el teléfono encendido (con todas las funcionalidades y tarjetas de comunicaciones apagadas) y por tenerlo exactamente igual pero asociado a un punto de acceso WLAN y en estado *idle*, es decir, sin enviar ni recibir datos. El estado *sleep* no tiene tanto interés puesto que no vamos a utilizar el modo de ahorro de energía. Los valores que conseguimos están afectados por la presencia de tráfico de difusión y *multicast* en nuestra red, y también por el ruido debido a otras redes en el mismo área de influencia aunque en menor medida. Esto también se aplica al resto de mediciones, a menos que las tomásemos en un lugar aislado, como podría ser dentro de un armario metálico. Puesto que queremos conseguir valores tan cercanos a la realidad como sea posible, nuestros experimentos no están aislados, y la red de nuestro escenario tiene una carga baja-media, sólo la debida a la operación del punto de acceso.

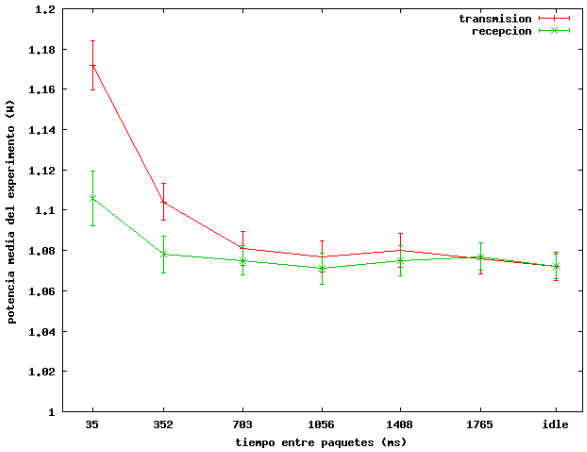


Fig. 3. Transmisión y recepción con tamaño de paquete fijo, cadencia variable.

Después, estudiamos la transmisión y la recepción dado un tamaño de paquete fijo de 600 bytes y variando la cadencia de envío, como puede verse en la Fig. 3. En ella, como en el resto de gráficas, se puede ver la media de consumo para cada caso,

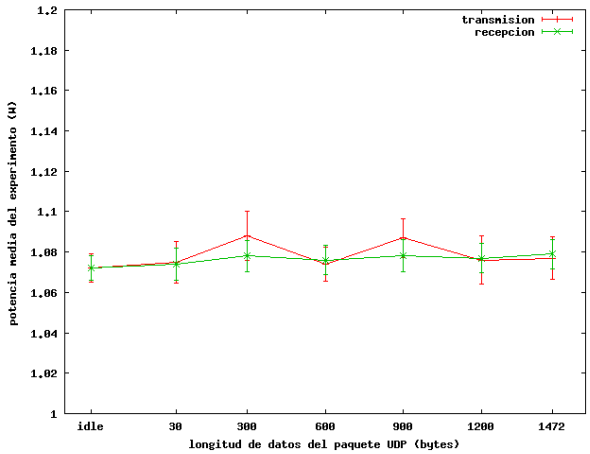


Fig. 4. Transmisión y recepción con cadencia fija, tamaño de paquete variable, a 11 Mbps.

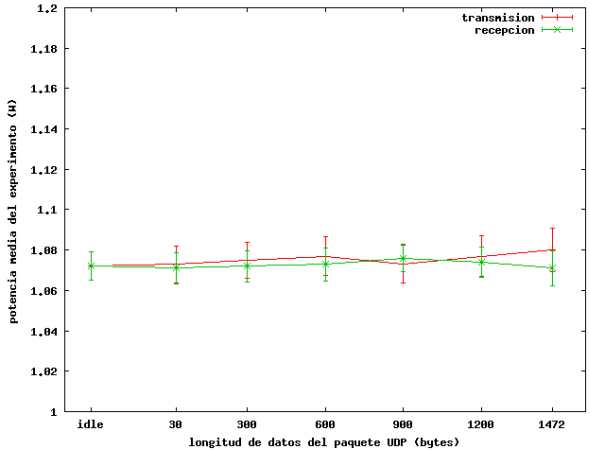


Fig. 5. Transmisión y recepción con cadencia fija, tamaño de paquete variable, a 1 Mbps.

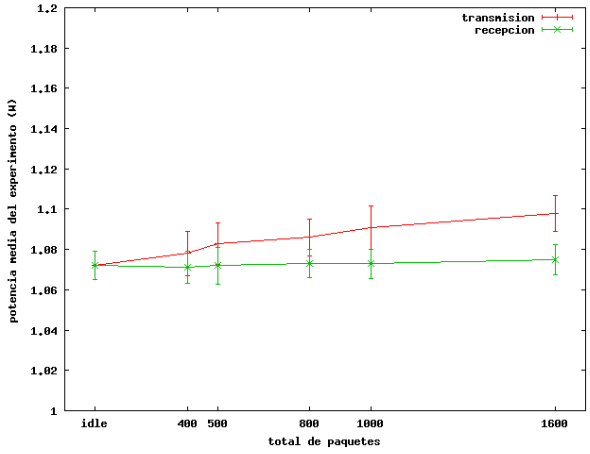


Fig. 6. Transmisión y recepción con tasa fija.

junto con la correspondiente desviación típica. Para intervalos entre paquetes muy cortos, el consumo sube drásticamente: casi 0.1 W más, cuando la transmisión consume alrededor de 0.09 W y la recepción menos de 0.08 W en el resto de casos.

Por otro lado, las pruebas con cadencia fija mostraron que el tamaño de paquete variable parece no tener influencia en el resultado, ya que el consumo fluctuaba en un rango entre 1.07 W y 1.09 W para todas las pruebas. Esto se puede ver en la Fig. 4. Repetimos las pruebas forzando en el punto de acceso una tasa de transmisión de 1 Mbps para alargar el tiempo durante el que se está enviando un paquete, éstas se muestran en la Fig. 5. Se puede apreciar una ligera pendiente ascendente en proporción con el tamaño de los datos UDP, sobre todo en el primer tramo, aunque la variación es tan pequeña que podría ser poco significativa.

Cuando tanto el tamaño de paquete como la cadencia se variaban para mantener una tasa fija, pudimos ver que el consumo de potencia medio crecía con el número total de paquetes enviados durante el experimento. Esto significa que la principal razón de gasto es el cambio entre el estado *idle* y la transmisión/recepción y/o viceversa. Estas pruebas se muestran en la Fig. 6.

E. Conclusiones

Hemos observado el consumo de potencia cambiando diferentes variables de una comunicación periódica: el intervalo entre paquetes, el tamaño de éstos, y el número de ellos manteniendo fija la tasa. Podemos extraer algunas conclusiones:

1) Hemos comprobado que, efectivamente, la transmisión es más costosa que la recepción. De hecho, los niveles de potencia necesarios para recibir son muy cercanos a los del estado *idle*. La literatura previa ya mostraba los mismos resultados.
2) Otro punto que hemos podido verificar es que los datos que hemos obtenido se acercan al modelo lineal ya propuesto anteriormente: el consumo varía tanto con el tamaño de los paquetes como con el número de ellos.
3) El resultado más sorprendente es que el tamaño de los paquetes parece influir poco en el consumo. Puesto que el tiempo durante el que se transmite el paquete es muy corto, el coste fijo asociado a la transmisión o recepción de un paquete tiene más peso, de ahí que notemos una dependencia mucho más clara del número de paquetes intercambiados. Esto nos da una pista clara sobre estrategias de ahorro a alto nivel.

Con los resultados obtenidos se plantean una serie de líneas de investigación interesantes, como es el desarrollo de un modelo matemático que permita predecir el consumo, así como definir estrategias para reducir el consumo energético en aplicaciones de red. En concreto, vamos a desarrollar este punto a continuación, haciendo un primer repaso sobre un conjunto de guías de diseño para aplicaciones UDP.

IV. Guías para reducir el consumo en aplicaciones sobre UDP

La RFC 5405, *Unicast UDP Usage Guidelines for Application Designers*, es el resultado del activo grupo de trabajo Transport Area (adoptado después por el Network WG) y quiere ofrecer consejos de valor para diseñadores de aplicaciones que pretendan usar UDP en el nivel de transporte.

El objetivo principal es apoyar el diseño de aplicaciones amigables con redes amplias, poniendo especial atención en los problemas derivados de la falta de control de congestión inherente a UDP. Los puntos cubiertos en la RFC son el control de congestión, el tamaño de paquete, la confiabilidad, temas relacionados con el *checksum*, el paso por *middleboxes*, programación, y temas relacionados con ICMP.

Aunque la prevención de congestión y la búsqueda de confiabilidad son de gran importancia, para redes inalámbricas formadas por dispostivos móviles es de igual importancia una buena gestión de la energía disponible. Podemos añadir algunos comentarios en este sentido a dos de los puntos tratados en la RFC: el tamaño del mensaje y el paso por *middleboxes*, dejando el resto para revisiones futuras.

A. Tamaño de paquete

Los autores de la RFC recomiendan no exceder la MTU (*Maximum Transfer Unit*) del camino para evitar la fragmentación. Puesto que el hecho de enviar un paquete es el mayor motivo de consumo de potencia para el teléfono, mucho más que el tamaño del mensaje, podríamos pensar que es mejor enviar un paquete del tamaño máximo que permita nuestro segmento de red. Sin embargo, la fragmentación en el camino puede traer problemas posteriores como la retransmisión de todo el paquete debido a fragmentos perdidos, por lo que es mejor evitarla desde el principio.

Teniendo este objetivo en mente, los autores presentan tres opciones:

- utilizar información sobre la MTU proveniente del nivel IP, esto es, limitar el tamaño al MMS_S (*maximum transport-layer message size for sending*), el cual es la MTU configurada para la interfaz de salida menos el tamaño de la cabecera IP (la MTU de 802.11 es 2312 bytes, pero normalmente se configura como 1500 bytes);
- implementar un mecanismo de descubrimiento de la PMTU (*Path MTU*);
- utilizar un tamaño de paquete inferior a la MTU efectiva para el envío por defecto (EMTU_S, *Effective MTU for Sending*), que es 576 bytes, si el destino no está en la red local.

Tanto la eficiencia de la red como el consumo de potencia mejoran cuanto más se acerca el tamaño del paquete al PMTU. Por tanto, el mecanismo de descubrimiento del PMTU sería la major opción. Sin embargo, implica un intercambio adicional de mensajes ICMP y retransmisiones hasta que su valor es finalmente definido. En el peor de los casos, sería necesario reajustar el tamaño del paquete tantas veces como el mínimo entre el número de segmentos en el camino menos uno (en el que está el dispositivo ya es conocido), y el número de MTUs diferentes en la red. Para un camino aleatorio en Internet, se espera encontrar no más de 30 saltos, siendo el caso más común entre 10 y 26 saltos, según [24]. En [25] hay una tabla con las 16 MTUs habituales en Internet según las opciones del nivel MAC. Sin embargo, varios autores consideran, como en [26], que el caso de MTU más típico es 1500 bytes puesto que la mayoría de las redes usan Ethernet. También existe la posibilidad de un cambio en el camino durante la transmisión, pero puede ser despreciada ya que dos tercios de los caminos en Internet son persistentes durante días o semanas, y el otro tercio cambia debido a modificaciones dentro de la misma red [27].

Con todo esto, podemos concluir lo siguiente: Primero, por lo que sabemos sobre la MTU típica, enviando paquetes de 1500 bytes es muy probable que no se fragmenten a lo largo del camino, pero no podemos estar seguros de ello. Implementar el descubrimiento de la PMTU garantizaría que no habrá fragmentación (suponiendo que el camino no va a cambiar), pero puede añadir un coste variable a la comunicación. Por último, utilizar paquetes de tamaño EMTU_S sería ineficiente debido a que se precisaría enviar un número más alto de paquetes, subiendo así el consumo. En la Fig. 7 podemos ver el punto correspondiente a enviar el número necesario de paquetes de 576 bytes (EMTU_S) a la misma tasa fija que los experimentos anteriores.

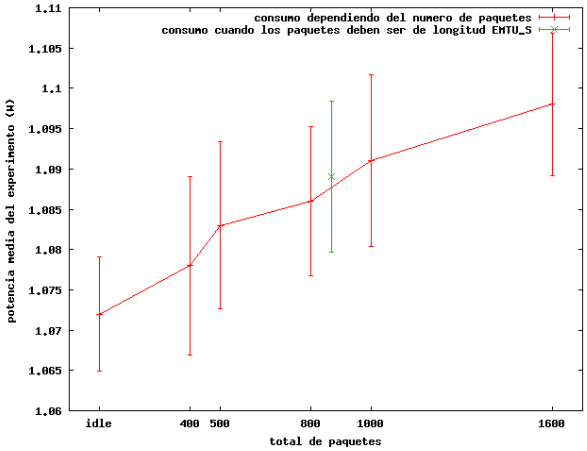


Fig. 7. Transmisión de 500 KB en 10 min cambiando el tamaño de los paquetes, también para paquetes de 576 B.

Por tanto, si es necesario mantener una cierta tasa de envío, consideramos que es mejor utilizar descubrimiento de PMTU. Si la transmisión es esporádica, un tamaño de 1500 bytes es una buena opción. El tamaño EMTU_S, aunque más seguro, podría significar un mayor gasto energético a largo plazo. El punto de inflexión entre utilizar un tamaño de paquete de 1500 bytes y descubrir el PMTU, sin embargo, es difícil de estimar y sería necesario realizar una investigación estadística para averiguar cuántos mensajes son necesarios hasta que se encuentra la PMTU.

B. Paso por middleboxes

Se entienden como *middleboxes* aquellos dispositivos, como los NATs o los cortafuegos, que separan parte de un camino del lado exterior de una red. Estos dispositivos comparten las características de bloquear paquetes que vienen de fuera si no pertenecen a un flujo registrado que tiene su comienzo en el interior. Si pasa el tiempo sin que haya intercambio de paquetes en dicho flujo, éste es eliminado y a partir de ese momento es imposible recibir nada del interlocutor en el exterior hasta que el flujo se reinicie desde el interior de la red. La RFC presenta las tres formas de manejar el problema en las aplicaciones:

- La aplicación, por su propia naturaleza, no se ve afectada por la destrucción del flujo. Un claro ejemplo de esto sería el esquema pregunta-respuesta de DNS.

- La aplicación es capaz de manejar las reconexiones de forma transparente para el usuario. Por ejemplo, las aplicaciones de correo estarían en este grupo.
- La aplicación se ve afectada por las reconexiones y debe evitarlas mediante el uso de *keep-alives*. Ejemplos de esto son las aplicaciones multimedia.

Los autores de la RFC recomiendan evitar el uso de *keep-alives* siempre que sea posible, ya que estos paquetes añaden carga a la red sin aportar datos nuevos. Desde el punto de vista energético, también suponen un gasto extra. Si es que la aplicación los necesita, la RFC aconseja enviarlos no más frecuentemente que una vez cada 15 segundos. En [28], un documento de Nokia, definen 20 segundos como intervalo óptimo dado el coste y el intervalo más seguro. En la Fig. 8 podemos ver cómo ambos intervalos sitúan el consumo en la región de la varianza del estado *idle*, lo que significa que los dos están lo suficientemente cerca del mínimo gasto de potencia posible.

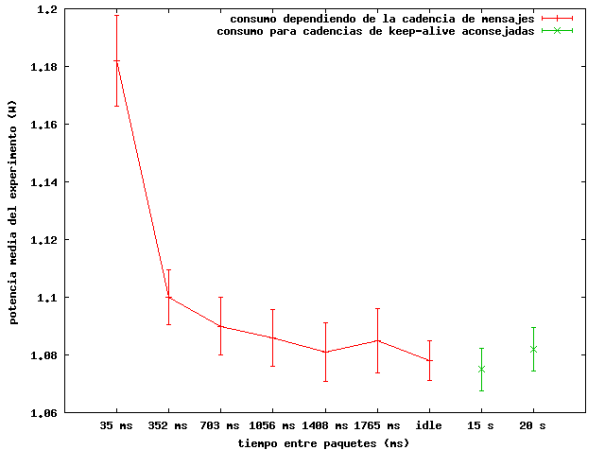


Fig. 8. Transmisión con tamaño de paquete fijo, también para intervalos de 15 y 20 segundos.

V. Conclusiones

Hemos descubierto aspectos importantes acerca de la utilización óptima de la interfaz de red, que nos pueden ayudar a desarrollar aplicaciones energéticamente eficientes para dispositivos móviles limitados e incluso hacer un mejor uso de los protocolos de transporte existentes. Aunque muchos trabajos de investigacion demuestran que el consumo de energía depende del tamaño del paquete, hemos observado que en la práctica este valor es prácticamente despreciable comparado con el coste fijo por paquete. Por tanto, una buena práctica en el diseño de aplicaciones en red sería enviar pocos paquetes grandes en vez de muchos paquetes más pequeños, al menos con potencias de transmisión bajas.

Todavía hay mucho trabajo que hacer en este área. Por ejemplo, queremos caracterizar completamente el consumo de dispositivos móviles durante comunicaciones IEEE 802.11 incluyendo comunicaciones *multicast*/*broadcast*. Con esto podremos definir un modelo matemático que ayude en el diseño de estrategias de ahorro. Sería posible también idear extensiones adaptativas para aplicaciones populares dependiendo de las necesidades del dispositivo. Otra línea de trabajo futuro a corto plazo será completar la revisión de la RFC 5405 mediante la evaluación de los otros aspectos presentados en él desde el punto de vista del consumo energético. Además, a partir de esto se podrían estudiar otros protocolos de transporte populares tales como TCP o SCTP. Finalmente, sería interesante averiguar cómo de precisas son las medidas que podemos tomar mediante herramienas *software* comparándolas con medidas físicas, puesto que últimamente este tipo de aplicaciones se está haciendo popular incluso para el usuario final.

Agradecimientos

Este trabajo ha sido parcialmente soportado por el proyecto Integración Vertical de Servicios Telemáticos de Apoyo al Aprendizaje en Entornos Residenciales Distribuidos, CCG08-UC3M/TIC-4479, y por Nokia Spain S.A. Agradecemos también a Sara Villanueva su valiosa aportación en la realización del estudio experimental.

Referencias

[1] G. Bosch Creus and M. Kuulusa, "Optimizing Mobile Software with Built-in Power Profiling," Book chapter in *Mobile Phone Programming* – Springer Netherlands, ISBN: 978-1-4020-5968-1, June 2007.

[2] P. Gauthier, D. Harada and M. Stemm, *Reducing Power Consumption for the Next Generation of PDAs: It's in the Network Interface!*, in Proceedings of the International Workshop on Mobile Multimedia Communications (MoMuC),1996.

[3] L. Feeney and M. Nilsson, *Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment*, in Proceedings of IEEE INFOCOM 2001.

[4] I. Martínez, C. Campo, C. García-Rubio, *Caracterización de Consumo Energético de Interfaces Inalámbricos sobre Linux*, Bachelor Work, University Carlos III of Madrid, 2007.

[5] IETF Network WG, *RFC5405 - Unicast UDP Usage Guidelines for Application Designers*, http://www.ietf.org/rfc/rfc5405.txt

[6] A. Zanella and F. De Pellegrini, *Mathematical Analysis of IEEE 802.11 Energy Efficiency*, in Proceedings of WPMC04, 2004.

[7] J. Lorchat and T. Noel, *Power Performance Comparison of Heterogeneous Wireless Network Interfaces*, Vehicular Technology Conference, 2003.

[8] J. Ebert, S. Aier, G. Kofahl, A. Becker, B. Burns and A. Wolisz, *Measurement and Simulation of the Energy Consumption of an WLAN Interface*, Technical Report TKN-02-010, Technical University Berlin, 2002.

[9] M. Gruteser, A. Jain, J. Deng, F. Zhao and D. Grunwald, *Exploiting Physical Layer Power Control Mechanisms in IEEE 802.11b Network Interfaces*, Technical Report, University of Colorado at Boulder, 2001.

[10] M. Stemm, P. Gauthier, D. Harada, Y. H. Katz, *Reducing Power Consumption of Network Interfaces in Hand-Held Devices*, IEICE Transactions on Communications, 1997.

[11] A. Cortes, C. García-Rubio, *Low-cost Power Measurement of PDA's 802.11b Network Interfaces*, Technical Report, University Carlos III of Madrid, 2006.

[12] IETF Network WG, *Extensible Peer Protocol (XPP)*, (draft-marocco-p2psip-xpp-02), May 2008.

[13] IETF Behave WG, *Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)*, (draft-ietf-behave-turn-15), June 2009.

[14] IETF IPFIX WG, *RFC5153 - IP Flow Information Export (IPFIX) Implementation Guidelines*, http://www.ietf.org/rfc/rfc5153.txt

[15] IETF MIPSHOP WG, *IEEE 802.21 Mobility Services Framework Design (MSFD)*, (draft-ietf-mipshop-mstp-solution-12), January 2009.

[16] R. Banginwar and E. Gorbatov, *Gibraltar: Application and Network Aware Adaptive Power Management for IEEE 802.11*, Proceedings of the Second Annual Conference on Wireless On-demand Network Systems and Services (WONS'05), 2005.

[17] E. S. C. Takahashi, *Application Aware Scheduling for Power Management on IEEE 802.11*, Performance, Computing, and Communications Conference (IPCCC '00), Conference Proceeding of the IEEE International, 2000.

[18] Y. He, R. Yuan, X. Ma, J. Li, C. Wang, *Scheduled PSM for Minimizing Energy in Wireless LANs*, IEEE International Conference on Network Protocols (ICNP 2007), 2007.

[19] J. Lorch, *A Complete Picture of the Energy Consumption of a Portable Computer*, Computer Science, University of California at Berkeley, 1995.
[20] acbPowerMeter, http://www.acbpocketsoft.com/Products/acbPowerMeter/acbPowerMeter-Overview-2.html
[21] acbTaskMan, http://www.acbpocketsoft.com/Products/acbTaskMan/acbTaskMan-Overview-7.html
[22] Nokia Carbide and Performance Investigator, http://www.forum.nokia.com/main/resources/tools_and_sdks/carbide_cpp/
[23] Nokia Energy Profiler, http://www.forum.nokia.com/main/resources/user_experience/power_management/nokia_energy_profiler/
[24] G. Armitage, C. Javier, S. Zander, *Post-game Estimation of Game Client RTT and Hop Count Distributions*, Netgames'06, October 2006.
[25] IETF Network WG, *RFC1191 - Path MTU Discovery*, http://www.ietf.org/rfc/rfc1191.txt
[26] IETF Network WG, *RFC 3449 - TCP Performance Implications of Network Path Asymmetry*, http://www.ietf.org/rfc/rfc3449.txt
[27] V. Paxson, *End-to-end Routing Behavior in the Internet*, Proceedings of the ACM SIGCOMM Conference, 1996.
[28] Forum Nokia, *Recommendations for Reducing Power Consumption of Always-on Applications*, Version 1.0, September 2007.

Caracterización del Perfil de Consumo de Energía de Servicios IP sobre teléfonos móviles

Almudena Díaz Zayas, Pedro Merino Gómez
Departamento de Lenguajes y Ciencias de la Computación,
Universidad de Málaga, Campus Teatinos, 29071, Málaga
almudiaz@lcc.uma.es, pedro@lcc.uma.es

***Resumen*—La capacidad de procesamiento y de memoria de los teléfono móviles se ha visto incrementada sustancialmente en los últimos años, así como el número de tecnologías inalámbricas incorporadas en dichos dispositivos (Infrarrojos, Bluetooth, GSM, UMTS, WiMax, NFC). De esta forma se han posicionado como dispositivos ampliamente usados para la navegación y acceso a los contenidos multimedia disponibles en Internet. Sin embargo el acceso a dichos contenidos supone un fuerte aumento del consumo de batería, recurso escaso en todo dispositivo móvil. En este artículo se realiza un análisis del consumo de energía en teléfonos móviles y se determinan los factores que provocan su aumento y que están estrictamente relacionados con el tráfico de datos cursado a través de las distintas tecnologías de acceso.**

***Palabras Clave*—Consumo, energía, teléfonos móviles, tráfico de datos, protocolos**

I. Introducción

En anteriores trabajos de los autores de este artículo se ha abordado el análisis del rendimiento de servicios específicos sobre redes celulares [9] [10] a nivel de red, transporte y aplicación. En este artículo nos centramos en el estudio del consumo de energía generado por aplicaciones y protocolos de diferentes perfiles y comportamientos. El objetivo del estudio es el diseño de las estrategias de roaming que se emplearán en la implementación del middleware propuesto en [7]. Dicho middleware tiene por objetivo permitir una gestión transparente de las tecnologías de acceso radio presentes en un terminal móvil en función del entorno y del grado de movilidad, siendo una parte importante las políticas de gestión del consumo de batería.

Existen en la literatura numerosos estudios sobre el consumo en dispositivos móviles. Muchos de ellos como [3] están centrados en pruebas de laboratorio con terminales diseñados específicamente para tal fin que tienen por objetivo estudiar el consumo de batería de los distintos elementos hardware que incorpora el terminal, como la pantalla, la memoria, el procesador, el auricular, etc. Dichos estudios son necesarios para optimizar el diseño de dispositivos móviles, sin embargo no son aplicables en el caso de estudio en el que nos encontramos por no considerar aspectos relacionados con las conexiones de datos. Otros artículos presentan estudios parciales que se centran en aspectos concretos como en el consumo de las interfaces WLAN [4], el estudio de un cierto tipo de aplicación [11] o factores relacionados con la gestión de los recursos radio que lleva a cabo el operador móvil y el uso de NAT y firewalls [6]. En [12] se lleva a cabo un estudio del consumo de potencia para la aplicación Mobile YouTube, pero no llega a concretarse la relación existente entre el tráfico cursado por la aplicación y el consumo. En [8] se lleva a cabo un análisis exhaustivo sobre el consumo de tarjetas de datos WLAN y 3G sobre un ordenador convencional. Las pruebas realizadas en este artículo se han llevado a cabo directamente sobre un dispositivo móvil. Una importante diferencia entre ambos escenarios es la pila de protocolos disponible en ambos tipos de dispositivos. La pila de protocolos del terminal móvil implementa una serie de RFCs específicas para entornos celulares, mientras que la pila de protocolos implementada en el ordenador presenta la configuración tradicional para redes fijas. Dado que en este artículo se analiza el impacto del tráfico de datos a través de distintos interfaces la configuración de la pila de protocolos debe ser tenida en cuenta como parte de un escenario de pruebas realista. Por otro lado en dicho artículo no se profundiza en el origen del consumo asociado a la transmisión de datos a través de interfaces 3G, sino que se ahonda más en el estudio de interfaces WLAN.

En este artículo, a diferencia de los trabajos citados, se proporciona un análisis general de los distintos factores que afectan al consumo de energía asociado a una conexión de datos, desde la tecnología de acceso en uso, pasando por las aplicaciones y protocolos empleados, la gestión física del enlace y los fenómenos propios de las tecnologías celulares como los handovers. Se pretende, por tanto, proporcionar una visión global de los factores que afectan al consumo de energía debido al tráfico de datos y cuantificar el impacto de cada uno de dichos factores sobre el consumo total.

En el artículo se define un entorno real de pruebas usando terminales comerciales basados en la Serie 60 de Symbian. Esto nos permite evaluar el rendimiento energético de los protocolos implementados en un teléfono móvil, los cuales incorporan extensiones específicas para sistemas inalámbricos. Para dicho sistemas operativo se han implementado una serie de herramientas que nos permiten capturar el tráfico cursado así como medir el consumo de energía, los niveles de señal transmitida y recibida y el volumen de tráfico entrante y saliente para distintas tecnologías de acceso. Durante las medidas también se ha constatado que el consumo de memoria y la carga del procesador son similares para las distintas aplicaciones evaluadas. Se estudia también el consumo de los servicios de mensajería móvil MMS y SMS y cómo estos podrían se empleados conjuntamente con los protocolos IP para optimizar el consumo final de energía.

El artículo está organizado de la siguiente forma. En la sección 2 se lleva acabo la descripción de distintos escenarios en los que se han realizado medidas para proporcionar una visión global de los distintos factores que afectan al consumo de energía en un terminal móvil y la medida en la que lo hacen. En la sección 3 se introducen las medidas de consumo de energía asociado a las conexiones y al tráfico de

datos. Las medidas han sido realizadas con diferentes tipos de aplicaciones. Asimismo se procede a la identificación de los factores que influyen sobre el consumo asociado a los distintos tipos de flujos de datos. En la sección 4 se aplican técnicas de regresión lineal para cuantificar el impacto de los factores identificados en los apartados previos. Finalmente en la sección 5 se presenta un resumen de los resultados obtenidos.

II. Escenarios de referencia

En este apartado se caracterizan un conjunto básico de escenarios de funcionamiento que servirán de referencia para comparar las medidas de consumo de energía relacionadas con el tráfico de datos. En primer lugar se describen las herramientas empleadas para realizar las mediciones y las funcionalidades que proporcionan. A continuación se procede a la descripción de los distintos escenarios de referencia en los cuales se identifican estados básicos de funcionamiento del terminal y se analiza el consumo para cada uno de ellos.

A. Herramientas de medidas

Las medidas se han llevado a cabo sobre el terminal Nokia 5800 Xpress Music. Dicho terminal está basado en la plataforma Serie60 de Symbian OS. El terminal está equipado con una sola CPU del tipo ARM 11 con una frecuencia de 368 MHz y dispone también de 128 MB de memoria SDRAM. Para la monitorización del consumo de energía en tiempo real se ha usado la aplicación Nokia Energy Profile (NEP) desarrollada por Nokia. Para la captura del tráfico de datos y la monitorización de la tecnología de acceso en uso se ha usado la herramienta SymPA [1], desarrollada por los autores de este artículo. Por último se ha implementado una aplicación software que permite unificar los sellos de tiempo de ambas aplicaciones de forma que las medidas de tráfico de SymPA pueden ser correladas con las medidas realizadas por la herramienta NEP. Dicha aplicación está basada en las APIs externas proporcionadas por Nokia para acceder a las medidas del NEP.

B. Caracterización del consumo de energía en el escenario de referencia

Se han identificado 4 casos de uso básicos del terminal y se ha procedido a la caracterización del consumo de energía en cada uno de ellos: terminal en modo de espera, terminal con pantalla activa, herramientas de medida en ejecución y llamada de voz. En los siguientes apartados se definen las condiciones de funcionamiento en cada uno de los casos de uso identificados y los valores de consumo obtenidos para cada uno de ellos. Las medidas, para cada uno de los casos, se han extendido durante un periodo de 9 minutos.

1) Terminal en modo de espera: En el modo de espera la pantalla permanece totalmente apagada y no existe ninguna aplicación en ejecución. La energía total acumulada durante el intervalo de medidas seleccionado en el modo de espera es de 0,001 Ah (amperios hora) y la potencia media consumida es de 0,02994 W. La potencia media de la señal recibida es de -57 dBm. En dicho modo no existe potencia en transmisión.

2) Terminal en modo activo: Se considera el "modo activo" como aquel modo en el cual la pantalla del terminal permanece iluminada. La iluminación de la pantalla es un de las principales fuentes de consumo en un terminal móvil. Los terminales suelen incorporar un temporizador que permite desactivar la iluminación de la pantalla tras un cierto periodo de inactividad. Para evitar que el consumo debido a la pantalla pueda alterar las medidas realizadas se ha configurado el brillo de la pantalla al mínimo y se ha deshabilitado el temporizador de desactivación de la iluminación de la pantalla, de forma que el consumo debido a la pantalla se mantiene constante durante todo el periodo de medición. Durante los intervalos de medida seleccionado el consumo total de energía asciende a 0,016 Ah, y el consumo medio de potencia instantánea se sitúa entorno a los 0,3911 W. La potencia de señal recibida es de -66,98 dBm. No existe potencia en transmisión.

3) Consumo de las herramientas de medida: Para evaluar el efecto sobre el consumo de energía de las herramientas de medidas empleadas durante las pruebas realizadas se ha observado, durante el mismo intervalo de tiempo, el consumo de energía que supone la ejecución de dichas herramientas con la pantalla apagada. Como vemos en la figura 1 el incremento en el consumo de energía que supone la ejecución de dichas herramientas es despreciable.

4) Llamadas de voz: La figura 1 muestra la energía acumulada para cada uno de los modos de funcionamiento caracterizados en este apartado. En la gráfica se puede apreciar que, dentro de los casos de uso considerados, la llamada presentan un consumo de energía considerablemente mayor. Se ha tomado como referencia una llamada de voz 3G de 9 minutos de duración. El consumo de energía acumulado corresponde a 0,051 Ah. La media de la potencia instantánea durante dicho intervalo es de 1,24 W y los niveles de transmisión y recepción se mantienen en torno a los -26 y -58 dBm respectivamente.

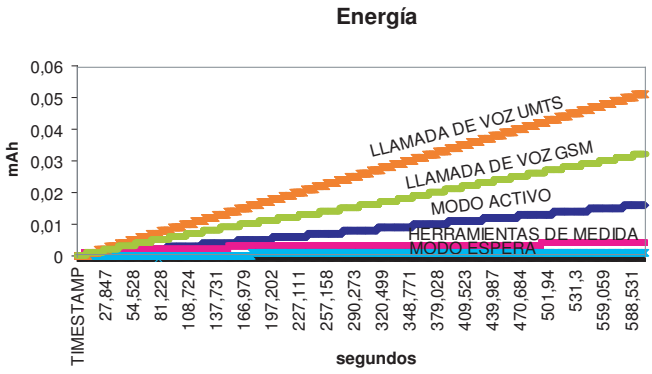


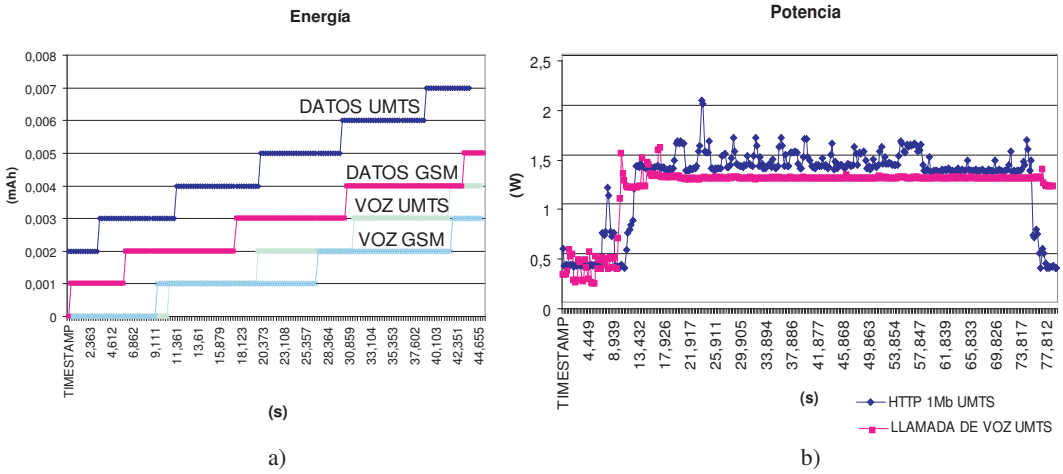
Fig. 1. Comparativa consumo de energía: modo espera, modo activo, herramientas de medida y llamadas de voz

III. Medidas de consumo de energía en terminales móviles

En este apartado se analizan los distintos factores relacionados con las conexiones de datos que afectan al consumo de energía: tecnologías de acceso, protocolos y configuración de la pila de protocolos disponible en el terminal, perfil del tráfico y configuración de las portadoras de datos del operador móvil.

	HSDPA	UMTS	GPRS	WLAN
Duración (s)	18,986	42	205,197	6,997
Potencia consumida (W)	1,7812	1,503	1,07	1,204
Energía (Ah)	0,004	0,007	0,017	0,002
Enlace descendente (bytes/s)	55062,733	25899,244	5392,759	143220,48
Enlace ascendente (bytes/s	1411,146	709,524	209,008	2066,88
Señal recibida (dBm)	-67,026	-65,398	-68,59	N/A
Señal transmitida (dBm)	-24,946	-24,881	N/A	N/A

Tabla I
COMPARATIVA DESCARGA HTTP USANDO DISTINTAS TECNOLOGÍAS DE ACCESO



a) b)

Fig. 2. Comparativa de consumo entre llamada de voz y conexión de datos en UMTS

También se estudiará el impacto de los cambios de celda y los handovers entre GSM y UMTS/HSDPA en el consumo de energía.

A. Tecnologías de acceso

Para llevar a cabo un estudio comparativo de las distintas tecnologías de acceso radio presentes en un terminal móvil se ha tomado como referencia una aplicación de navegación Web basada en el protocolo de aplicación HTTP y se ha procedido a la descarga de un fichero de 1 MByte a través de WiFi, GPRS, UMTS y HSDPA. En la tabla I se muestra la duración total de la descarga, los valores medios de potencia, tasa de descarga, tasa de envío, potencia de señal recibida y transmitida y la energía total acumulada durante la descarga para cada una de las tecnologías. En el apartado de tecnologías celulares los mejores resultados en relación al consumo de energía se obtienen para HSDPA. El consumo de potencia instantáneo se incrementa en un 66% en comparación con GPRS, sin embargo el tiempo de transferencia se ve reducido en un 90%, resultando finalmente en una reducción del 76% de la energía consumida. En el caso de WiFi el consumo de energía acumulada se ve reducida a la mitad respecto a HSDPA. Para WiFi obtenemos los menores tiempos de transferencia (presenta una reducción del 36% respecto a los valores obtenidos en HSDPA) así como el menor consumo de potencia instantánea, convirtiéndose así, en la tecnología más eficiente tanto en términos de velocidades de transferencia como en términos de consumo.

Si comparamos los datos de la tabla I con los consumos de la gráfica 2 a) vemos como tanto en GPRS como en UMTS el consumo de una conexión de datos es mayor que el de una llamada de voz. En la figura 2 a) vemos que existe una diferencia de 0,001 Ah en GSM y una diferencia de 0,002 Ah en UMTS. Para encontrar la raíz de estas diferencias en el consumo se ha presentado en la figura 2 b) el consumo de potencia instantáneo a lo largo de una llamada de voz y de una conexión de datos Http. Mientras que el consumo de potencia se mantiene constante a lo largo de la llamada de voz en la conexión de datos existen fluctuaciones que llegan a alcanzar el 30% respecto del consumo medio. Por tanto para conexiones de datos existe una componente del consumo que depende del tráfico.

En el siguiente apartado se profundiza en los distintos factores que contribuyen al consumo de energía en una conexión de datos UMTS.

B. Consumo asociado a una conexión de datos UMTS

En el caso de UMTS la principal fuente de consumo de energía se asocia al protocolo de control de recursos radio (RRC). En dicho protocolo se definen los estados de las portadoras de radio sobre las que se establecen las conexiones de datos. Se definen así los estados Cell_DCH, Cell_FACH, Cell_PCH, URA_PCH e IDLE (ver figura 3). En el estado Cell_DCH se le dedican al terminal de usuario un canal físico de bajada y otro de subida. Este estado está orientado a la transferencia de grandes cantidades de datos a tasas de hasta 2 Mbps a costa de aumentar el consumo de energía. En el estado Cell_FACH el terminal comparte canal tanto para el enlace descendente como el ascendente. Esta orientado al envío de pequeñas cantidades de datos. En el estado Cell_PCH

el terminal no puede enviar ni recibir datos, puede ser avisado de que hay datos para él, en el caso de recibir datos conmutará a algunos de los dos estados anteriores. En el estado IDLE no existe conexión RRC aunque el contexto PDP (Packet Data Protocol) puede permanecer abierto y el terminal puede ser avisado de datos entrantes o salientes a través del canal de control, es decir el terminal puede ser localizado a través de la dirección IP que le fue asignada durante el establecimiento de la conexión. Para un diseño óptimo de la temporización de las transiciones entre los distintos estados es necesario llegar a una solución de compromiso entre el consumo de potencia y los tiempos de respuesta. Si se conmuta rápidamente a los estados de menor consumo de potencia se optimiza el consumo de energía, sin embargo se empeora la latencia de la conexión puesto que los canales de transporte deben volver a asignarse o incluso se debe restablecer la conexión RRC.

En la figura 4 se puede ver la transición entre los distintos estados, la duración de estos, el consumo de potencia, los niveles de transmisión y recepción, y el tráfico cursado en cada uno de los estados. Para la celda de Vodafone en la que se ubicaban las pruebas se lleva a cabo una transición directa entre el estado CELL_DCH al estado IDLE 20 segundos después de recibir los últimos datos, mientras que para Movistar se produce la transición al estado CELL_FACH 10 segundos después de recibir los últimos datos y permanece en este estado 10 segundos antes de que se produzca la transición al estado IDLE. Como vemos en la figura 4 la recepción de una pequeña cantidad de datos produce la transición al estado CELL_DCH y un incremento de 1W en el consumo de potencia que se mantiene constante en este valor hasta que se vuelve al estado IDLE. Este análisis nos ayuda a entender el comportamiento que experimenta la potencia instantánea durante la descarga de datos HTTP representada en la figura 2b). Tras la correlación de los datos de potencia y el tráfico capturado se ha comprobado que las fluctuaciones de la potencia coinciden con periodo de tiempo en el cual se lleva a cabo la transferencia HTTP y que el intervalo final en el que la potencia se estabiliza se corresponde con el tiempo que tarda en conmutar al estado IDLE, durante el cual no hay tráfico.

En general los valores de los temporizadores configurados en las redes de Vodafone y Movistar en España son bastante elevados respecto de los 2 segundos recomendados en [6]. En base a los datos obtenidos podemos concretar que el consumo de energía tiene asociada una componente, independiente del tráfico cursado, que depende de los estados RRC y de la duración de estos y una componente que depende del volumen de tráfico cursado.

1) NAT Y Firewalls: Durante las pruebas se han detectado un gran número de conexiones intentando acceder a puertos donde existen vulnerabilidades conocidas en sistemas operativos tradicionales. EstO es debido a que las redes en las cuales se han realizado las pruebas no disponen de NAT o firewalls. El impacto de estos elementos de red sobre el consumo de potencia está ampliamente documentado en [6]. Sin embargo la ausencia de dichos elementos tiene también fuertes implicaciones en lo que respecta al consumo de potencia. El tráfico asociado a cada uno de estos ataques no es elevado pero supone una transición al estado de mayor

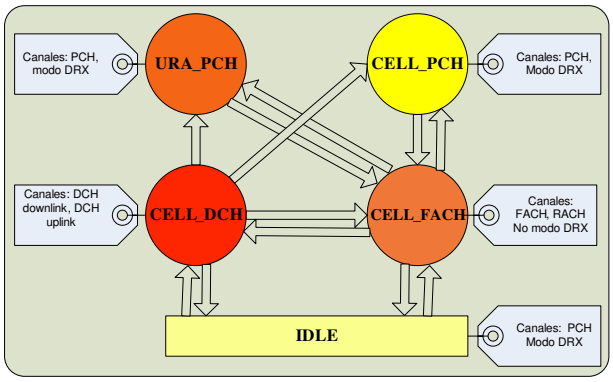


Fig. 3. Estados protocolo RRC durante una conexión de datos UMTS

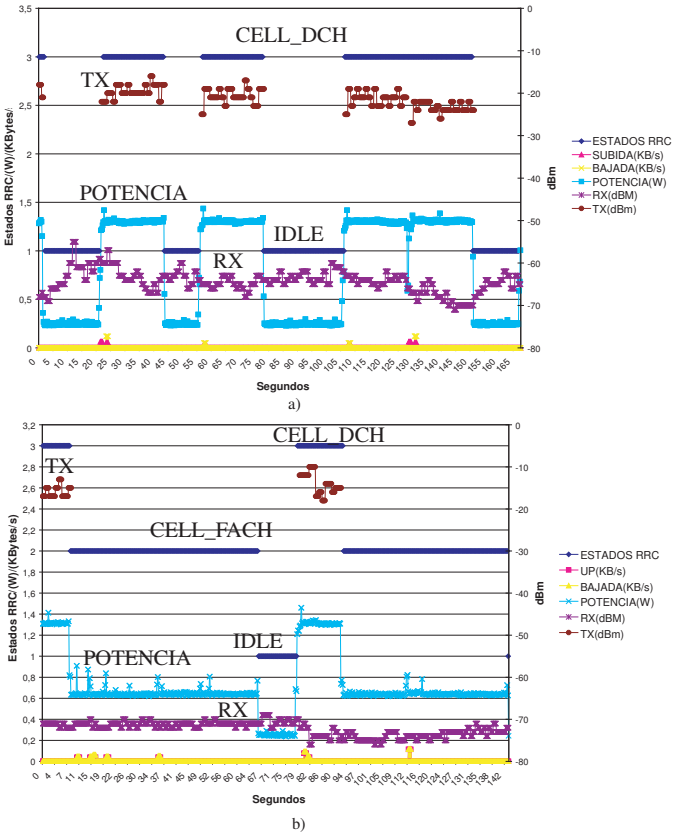


Fig. 4. Comparativa estados RRC, potencia consumida y volumen de tráfico entrante y saliente, a) Operador 1 b) Operador 2

consumo durante un periodo de tiempo que depende de la configuración RRC del operador, y que durante las pruebas se ha situado entre 10 y 20 segundos. Por tanto, en el caso de terminales móviles estos ataques no sólo suponen un riesgo de seguridad sino que supone un fuerte incremento en el consumo de energía.

C. Tipo de tráfico y protocolos

1) Aplicaciones de transferencia de datos: Con la aparición de las tarifas planas de datos se está incentivando desde los operadores la descarga de contenidos multimedia directamente al dispositivo móvil. En este apartado se evalúan dos conocidos protocolos de aplicación que pueden ser usados para la descarga de dichos contenidos. Los protocolos bajo estudio son HTTP y FTP. Para comparar el rendimiento

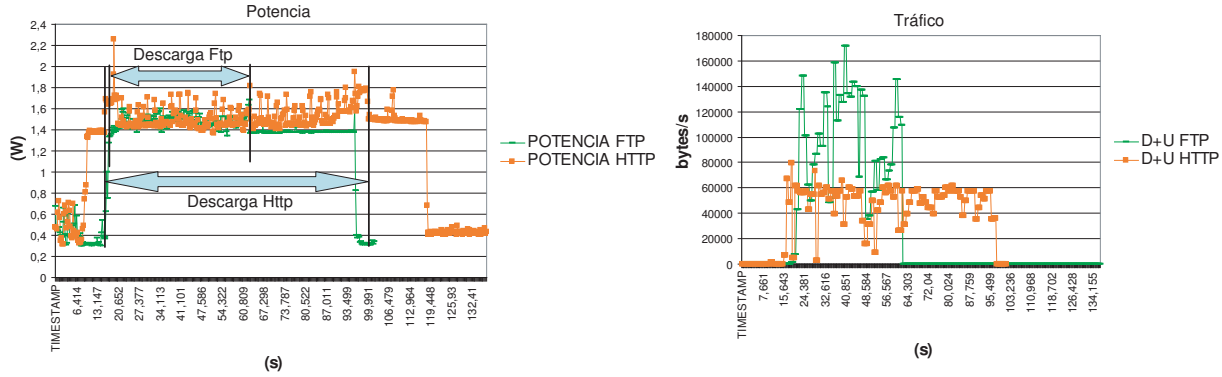


Fig. 5. Evolución de la potencia instantánea consumida a lo largo de una descarga HTTP y FTP.

de ambos protocolos se ha procedido a la descarga de un archivo de música de 3,56 MB desde una página Web y desde un servidor FTP. Los resultados obtenidos se muestran en la figura 5. Para el caso del protocolo HTTP el tiempo de descarga se duplica, duplicándose, también, la energía consumida.

La reducción del ancho de banda del protocolo HTTP se debe a la activación del flag PUSH en los paquetes TCP enviados durante la transferencia. En protocolos interactivos es muy común el uso de este flag que provoca el envío de los datos disponibles en el buffer de salida de forma inmediata. Durante el análisis de las trazas de tráfico se ha comprobado que el uso de este flag provoca la reducción del tamaño de los paquetes, que se traduce en una fuerte penalización en lo que respecta al ancho de banda debido al aumento de la sobrecarga de los paquetes de datos. La reducción del rendimiento del protocolo HTTP sobre redes celulares es un hecho documentado ampliamente en la literatura, el dato que realmente aporta la serie de experimentos llevados acabo en este apartado es la caracterización del comportamiento de la componente multiplicativa asociada al tráfico de datos. Dicho comportamiento puede apreciarse en la figura 5. Para un pequeño volumen de datos inicial se produce un incremento sustancial del consumo de potencia asociado al cambio de estado RRC. Una vez iniciado el intercambio de datos la potencia consumida varía conforme al volumen de tráfico cursado. En la sección 4 se obtendrá cuantitativamente el grado de correlación existente entre el tráfico y el consumo, lo que permitirá determinar el factor predominante en el consumo de energía.

2) Aplicaciones de tiempo real: En este apartado se ha elegido como aplicación un cliente de video streaming basado en el protocolo RTP sobre UDP para el transporte de video y en los protocolos RTSP y RTCP para el control de flujo y el envío de la información de control respectivamente. La ejecución del cliente de streaming tiene unos costes de consumo asociados a la reproducción del video. Para aislar el consumo debido únicamente al tráfico de datos se ha utilizado una aplicación que implementa los protocolos citados para intercambiar información con el servidor de video streaming pero que no reproduce el video recibido.

Para este caso la evolución del consumo de potencia a lo largo de la sesión de video streaming no experimenta el rizado apreciado en las anteriores aplicaciones. El comportamiento de la potencia es prácticamente plano.

En los siguientes apartados se analizarán otras aplicaciones con distintos perfiles.

3) Aplicaciones "always-on": En esta section se somete a evaluación el consumo de energía producido por el tráfico generado por el conocido cliente de VoIP, Skype. En la figura 6 se representan los niveles de potencia instantánea, energía, los niveles de la señal recibida y transmitida y el tráfico entrante y saliente. Durante nuestro estudio se ha encontrado que el consumo de potencia instantánea presenta 3 niveles claramente diferenciados. Los periodos de mayor consumo de potencia están asociados a periodos de transmisión y recepción en el estado CELL_DCH . Durante este intervalo el consumo instantáneo de potencia se sitúa entorno a 1,2 Watios. Tras conmutar al estado CELL_FACH el consumo decrece y se mantiene entorno a los 0,6 Watios. En este periodo se aprecia también tránsito de paquetes. En dicho estado existe un canal lógico dedicado y un canal físico de transporte común. También se puede apreciar que en dicho estado las variaciones en el consumo de potencia está fuertemente correladas con el tráfico cursado. Cuando se alcanza el estado IDLE la potencia consumida se sitúa en 0,2 Watios ya que la conexión RRC es liberada.

El volumen de tráfico cursado durante los 14 minutos que dura la ejecución de la aplicación es de 3.428 bytes y el tamaño medio de los paquetes intercambiados es de 100 bytes. Cada vez que se produce un intercambio de paquetes el consumo de potencia se incrementa en 1 Watios. Es decir pequeños picos de tráfico suponen un fuerte incremento en el consumo de potencia instantánea, lo que convierten a este protocolo en un protocolo extremadamente ineficiente en términos de energía.

Para aplicaciones "always-on" se deberían aplicar mecanismos alternativos que permitieran a las aplicaciones seguir "en contacto" de forma continuada sin una penalización tan fuerte en lo que respecta al consumo de energía. Por este motivo se analiza en el siguiente apartado el rendimiento en términos de energía de los servicios de mensajería SMS y MMS.

Por otro lado el efecto del tráfico no deseado comentado en el apartado anterior es especialmente nocivo para este tipo de aplicaciones ya que el contexto PDP permanece abierto durante largos periodos de tiempo, recibiéndose por tanto un gran número de ataques que incrementan el consumo.

4) Servicios de Mensajería Móvil: En este apartado se estudia el consumo que supone el envío de mensajes SMS

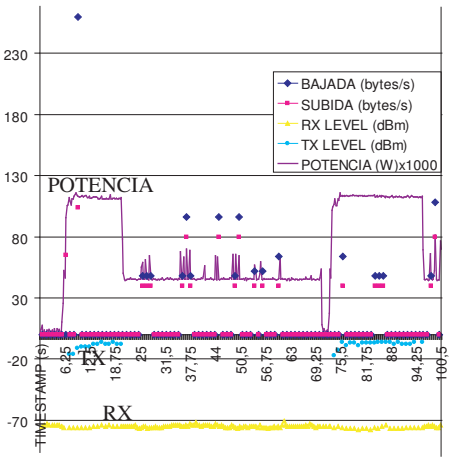


Fig. 6. Incrementos en el consumo de potencia instantánea asociadas a los niveles de tx y rx

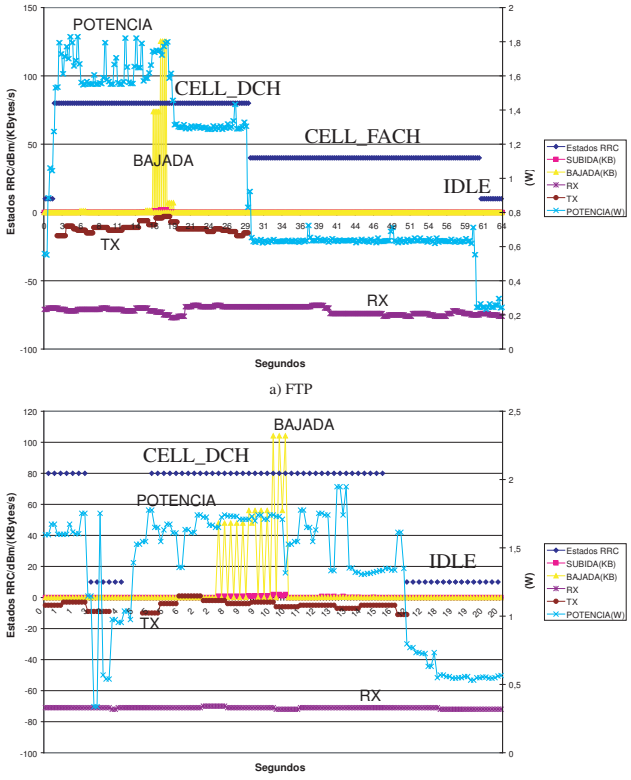
y MMS. Como se puede comprobar en la figura 7 el servicio de mensajería móvil resulta muchos más eficiente que por ejemplo una transferencia FTP del mismo tamaño, 200kB. Como se puede ver en la gráfica el servicio FTP resulta menos eficiente en términos de energía aunque el tiempo de transferencia sea menor. Para el caso de MMS se produce la transición al estado de menor consumo de potencia justamente al finalizar la receptión/transmisión de datos, mientras que para FTP una vez finalizada la transmisión no se produce la transición al estado de menor consumo hasta después de 10 segundos.

De igual forma, el consumo de potencia asociado al envío de un SMS de tamaño 1kB es de 1,2 W durante tan sólo 1 segundo aproximadamente. Por tanto dicho servicio se presenta como una solución energéticamente optima para ser utilizada, paralelamente con los protocolos de comunicación IP, en la implementación de aplicaciones ”always-on” en las que es necesario enviar y recibir periódicamente paquetes de tamaño pequeño. Otra mejora que se consigue con el uso de SMS es la disminución del tráfico no deseado, puesto que el contexto PDP no se debe mantener abierto durante largos periodos de tiempo.

D. Consumo de potencia asociado a los handovers

En este apartado la medidas se han llevado a cabo durante un trayecto en coche circulando por una autovía a una velocidad media de 80 km/h. En dicho escenario se mide el impacto en el consumo de los cambios de celda y del handover entre distintas tecnologías de acceso radio.

En la figura 8 se analiza el impacto de los handovers en el consumo de energía. Concretamente nos centramos en el comportamiento de los distintos parámetros medidos durante el segundo handover. En las inmediaciones de dicho handover se produce un incremento de la potencia consumida debido a que la señal de potencia de señal transmitida se aumenta para obtener una buena relación señal a ruido. Se producen, también, picos de tráfico, como se puede ver con el zoom de la 8 b) en el cual se representa el volumen de tráfico recibido, que coincide con los incrementos de la potencia instantánea.



a) FTP

b) MMS

Fig. 7. Estados RRC, potencia, niveles de señal transmitida y recibida y volumen de tráfico entrante y saliente a)FTP b)MMS

Justo después de este incremento se produce un fuerte decremento en el consumo de potencia asociada a una interrupción en el flujo de datos recibidos que se puede apreciar en la figura 8 c): interrupción de la tasa de bits de llegada, aumento del jitter y pérdidas de paquetes (flechas negras) detectadas después de cada handover. A pesar de aumentar la potencia de señal transmitida no se ha conseguido una relación señal a ruido que permitiera seguir adelante con la conexión de datos. En la figura 8 d) se puede ver las transiciones entre los distintos estado RRC en las inmediaciones del handover, y cómo los niveles de menor consumo de energía coinciden con las transiciones al estado IDLE de menor consumo.

A continuación se vuelve a apreciar el mismo comportamiento debido a un nuevo handover, primero se produce un incremento de la potencia y más tarde se produce una disminución. El mismo comportamiento se aprecia cuando se produce un handover entre HSDPA y GPRS. Sin embargo, como se puede ver en el primer handover en el cual se pasa de GPRS a UMTS primero y más tarde a HSDPA el consumo de potencia no experimenta dicho comportamiento, tan solo un incremento lógico de la potencia cuando se encuentra conectado a UMTS debido a que para dicha tecnología de acceso el consumo de potencia instantánea es mayor.

En este caso la gráfica corresponde a una sesión de video streaming en la cual se usa el protocolo el RTP sobre UDP para transporte del video. En el caso de usar protocolos de transporte con mecanismos de retransmisión como TCP las pérdidas asociadas al cambio de celda supondrían un aumento del tráfico debido a las retransmisiones. Lo que,

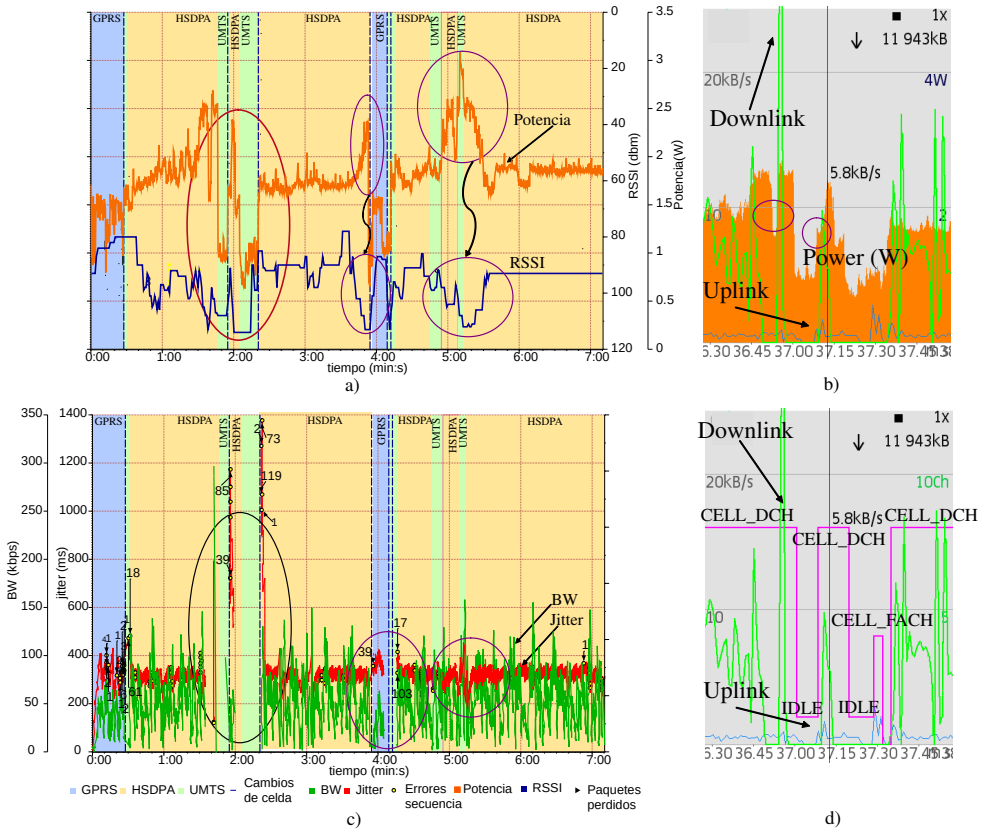


Fig. 8. Análisis del consumo de potencia durante cambios de celda

como hemos visto, implicaría un aumento del consumo debido al aumento del tráfico asociado a las retransmisiones. Pero también debido a la activación de técnicas de detección de congestión. Dichas técnicas provocan que el ancho de banda se reduzca, aumentando así la duración del intervalo de recepción y por tanto el consumo de energía. En este caso la pérdida de paquetes supondría una doble penalización en el consumo de energía.

Como hemos visto, en las inmediaciones de un handover, existen instantes de tiempo en los cuales se incrementa el consumo y otros en los cuales este disminuye. Para determinar cuantitativamente si el balance final del consumo es positivo o negativo se tienen que considerar, nuevamente, la temporización de las transiciones de los estado RRC, el tipo de protocolo empleado y otros factores como el tipo de handover, que no puede ser determinado sin tener acceso a la información de señalización del nivel de enlace. Sin embargo en base a la batería de pruebas realizadas usando como protocolo de transporte UDP se puede afirmar cualitativamente que el balance ha sido negativo, de forma que el handover entre distintas tecnologías se presenta como una solución óptima para reducir el consumo de energía en dispositivos móviles.

De igual forma durante las medidas realizadas se ha constatado que en entornos vehiculares el mayor consumo de potencia está relacionado con las variaciones en la potencia de la señal recibida que suelen estar asociadas a fenómenos que tienen lugar durante la propagación radio como el multicamino, los desvanecimientos etc. Este tipo de comportamiento se puede apreciar en la figura 8 alrededor del minuto 5.

IV. Análisis de los Resultados

En esta sección se analiza el tipo de relación existente entre el volumen de tráfico y el consumo de potencia. En la figura 9 se representa el resultado del análisis de regresión lineal llevado sobre uno de los múltiples experimentos realizados. Para los experimentos realizados se obtiene una componente constante de 1,4 Watios (P_o), que se asocia al estado RRC Cell_DCH. El coeficiente de correlación lineal P_1 es del orden de 10^{-6} Watios/bytes, lo que implica un aumento de 1 Watio por cada MB/s. Los factores de regresión lineal obtenidos son del mismo orden de magnitud que los obtenidos en [8] donde se analiza el coeficiente de regresión lineal pero no se hace referencia a la componente constante ni al origen de esta.

Para la cuantificación de la correlación se define el consumo de potencia instantánea como:

$$P(t) = P_o + P_1 \times r \tag{1}$$

Donde r es la tasa de transferencia en bytes/s. A continuación se define el consumo en términos de energía. Se define T como el intervalo de tiempo que tarda en transferirse X_B bytes a una tasa r:

$$T = \frac{X_B}{r} \tag{2}$$

$$E = P_o \times T + (P_1 \times r) \times T = P_o \times T + P_1 \times X_B \tag{3}$$

$$E(r) = X_B \times (\frac{P_o}{r} + P_1) \tag{4}$$

Finalmente llegamos a la conclusión de que para un volumen de tráfico X_B la energía consumida se puede minimizar aumentando la tasa de transferencia o lo que es lo mismo, disminuyendo el tiempo de transferencia. Si los valores obtenidos en los experimento realizados en el presente artículo fueran extrapolables a futuras tecnologías que proporcionen mayores tasas de transferencia, el consumo de energía se podría reducir hasta en un factor de 10, idealmente, al reducirse el peso relativo de la componente constante y predominar la componente dependiente del tráfico.

V. Conclusiones

Como hemos visto en el análisis del consumo de energía llevado acabo en este artículo, hay factores que dependen del operador y no pueden ser modificados por los diseñadores de aplicaciones móviles, sin embargo el conocimiento, por parte del desarrollador, de la gestión que se lleva acabo a nivel RRC le puede permitir adaptar los protocolos empleados para un mejor aprovechamiento global del consumo de energía.

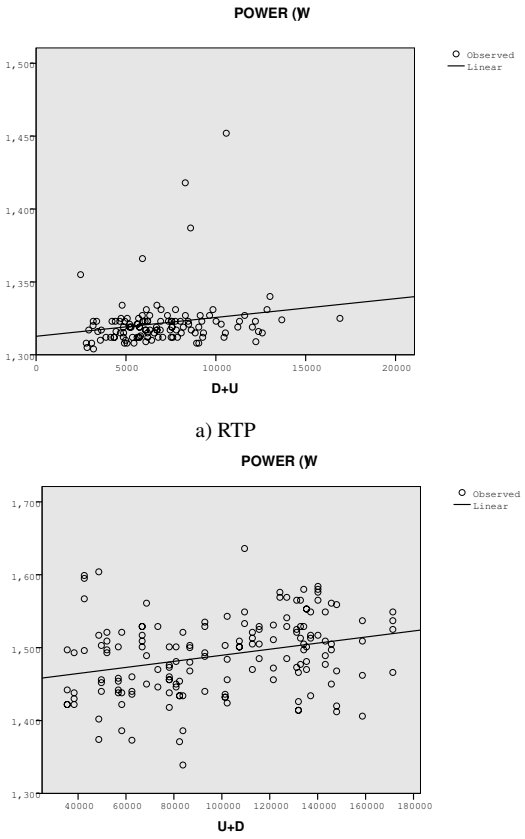
Por otro lado también se ha demostrado empíricamente, para las tecnologías de acceso evaluadas, que para optimizar el consumo de energía es necesario seleccionar en cada momento aquella que ofrezca una tasa de transferencia mayor. Por este motivo se ha evaluado también el impacto en el consumo de energía del handover entre GSM y UMTS/HSDPA, llegándose a la conclusión de que la ejecución de handovers resulta energéticamente eficiente y por tanto se puede recurrir a dicha técnica para seleccionar en cada momento la tecnología de acceso energéticamente más eficiente, sin que ello suponga penalización alguna en lo que respecta al consumo de energía.También se ha visto que las aplicaciones denominadas ”always-on” como Skype son las más afectadas por el dimensionado de los temporizadores de los estados RRC. El uso de temporizadores de corta duración permitiría optimizar el consumo de las aplicaciones ”always-on”, sin embargo aumentaría el retardo en aplicaciones interactivas, como por ejemplo las basadas en el protocolo HTTP. Por tal motivo se han evaluado otras alternativas llegándose a la conclusión de que el uso de servicios de mensajería móvil como SMS y MMS conjuntamente con protocolos IP permitiría reducir el consumo de energía en las aplicaciones ”always-on”.

Otro factor a tener en cuenta para optimizar el consumo de energía en dispositivos móviles es evitar el tráfico generado por ataques maliciosos ya que provoca una transición indeseada al estado RRC de mayor consumo.

Por ultimo para entornos vehiculares se producen incrementos notables del consumo de potencia instantánea asociados a disminuciones del nivel de potencia de señal recibida, mientras que los cambios de celda dentro de una misma tecnología de acceso radio no suponen un aumento de la energía consumida.

Agradecimientos

Este trabajo ha sido financiado por el proyecto nacional TIN 2005-09405-C02-01 y por el proyecto TIC 03131 de la Junta de Andalucía.



a) RTP

b) FTP

Fig. 9. a) Regresión lineal consumo/tráfico UDP b) Regresión lineal consumo/tráfico TCP

Referencias

[1] A. Diaz, P. Merino, F. J. Rivas, *Mobile Application Profiling for Connected Smartphones*, IEEE Pervasive Computing, (pendiente de publicación)

[2] T. Zhang, S. Madhani, P. Gurung, E. van den Berg *Reducing Energy Consumption on Mobile Devices with WiFi Interfaces*, IEEE Globecom 2005

[3] M.A. Viredaz, L. S. Brakmo, W. R. Hamburgen *Energy Managment on Handheld Devices*, ACM Queue, 2003

[4] M. Stemn, R. H. Katz *Measuring and Reducing Energy Consumption of Network Interfaces in Hand-Held Devices*, Transactions on Communications, Special Issue on Mobile Computing, vol. 8, 1997

[5] M.A. Viredaz, D.A. Wallach *Power Evaluation of a Handheld Computer*, IEEE Micro, 2003

[6] H. Haverinen, J. Siren, P. Eronen, *Energy Consumption of Always-On Applications in WCDMA Networks*, 65th Semi-Annual IEEE Vehicular Technology Conference (VTC 2007 Spring), 2007

[7] A. Diaz, P. Merino, *Towards a Lightweight Middleware for mobile multimedia communication awareness*, 4th International Mobile Multimedia Communications Conference, (MobiMedia), 2008

[8] K. Mahmud, M. Inoue, H. Murakami, M. Hasegawa, H. Morikawa *Energy Consumption Measurement of Wireless Interfaces in Multi-Service User Terminals for Heterogeneous Wireless Networks*, IEICE Transactions on communications, vol 88, n3, 2005

[9] A. Díaz-Zayas, P. Merino, L. Panizo, A.M. Recio, *Experimental analysis of peer-to-peer streaming in cellular networks*, in IEEE 21st International Conference on Advanced Information Networking and Applications (AINA-07), 2007

[10] A. Díaz, A. Gil, P. Merino, J. Muñoz, *x-AppMonitor µAgent: a tool for QoS measurements in cellular networks*, in 3rd International Symposium on Wireless Communication Systems, 2006

[11] A. Arjona, A. Yla-JMski, *VoIP Call Signaling Performance and Always-On Battery Consumption in HSDPA, WCDMA and Wifi*, International Conference on Wireless Communications, Networking and Mobile Computing (WiCom) 2007

[12] Yu Xiao, Ramya Sri Kalyanaraman, Antti Yla-Jaaski, *Energy Consumption of Mobile YouTube: Quantitative Measurement and Analysis*, Conference on Next Generation Mobile Applications, Services and Technologies, 2008

Un Módulo de Video-Conferencia para Moodle y una Experiencia Real de e-Learning en un Escenario Universitario

P. Manzanares-Lopez, J.P. Muñoz-Gea,
J. Malgosa-Sanahuja, J.C. Sanchez-Aarnoutse, J.J. Sanchez-Manzanares
Departamento Tecnologías de la Información y las Comunicaciones
Universidad Politécnica de Cartagena
Antiguo Cuartel de Antigones, Campus Muralla del Mar, C.P. 30202, Cartagena
{pilar.manzanares@, juanp.gea@,
josem.malgosa@, juanc.sanchez@, pepe.manzanares@si}upct.es

***Resumen*—En los últimos años, el desarrollo experimentado por las tecnologías de la información ha facilitado la implantación de plataformas de *e-learning* en el sistema universitario como solución alternativa o solución complementaria al método tradicional de enseñanza. En algunos casos, el *e-learning* es, en realidad, el único mecanismo proporcionado por algunas instituciones universitarias. En este artículo describiremos la experiencia de la Universidad Politécnica de Cartagena (UPCT) en la implantación y uso de dos plataformas diferentes de *e-learning*. La primera de ellas, llamada WebCT, corresponde a una solución de *e-learning* propietaria. Debido a los inconvenientes inherentes asociados al sofware propietario, recientemente la UPCT decició cambiar a una conocida plataforma de *e-learning* de libre distribución llamada Moodle. Además, este artículo presentará una contribución tecnológica al mundo del *e-learning*: un módulo para realizar audio/video conferencia en Moodle llamado Vicuc.**

***Palabras Clave*—e-learning, audio/video conferencia, moodle, desarrollo software**

I. Introducción

El desarrollo experimentado por las tecnologías de la información y las comunicaciones durante los últimos años ha modificado indudablemente los tradicionales escenarios educativos. La educación presencial puede ser complementada, o incluso sustituida, por educación a distancia. Este último tipo, la educación a distancia, ha evolucionado a través de diferentes etapas a lo largo de los años (cursos por correspondencia en papel, cursos en cassette, ...). En este ámbito, el *e-learning* podría verse como un último paso en la educación a distancia [1]. Podría definirse como un entorno educacional que engloba el uso de los recursos tecnológicos disponibles hoy en dia, incluyendo Internet, intranets, y cualquier tipo de dispositivos de audio y video. Sin embargo, el *e-learning* puede definirse de una forma más completa como un método educacional que proporciona contenidos empleando las nuevas tecnologías, al mismo tiempo que crea un proceso de enseñanza-aprendizaje que requiere de la integración de recursos y contenidos, de la interacción de todos los miembros participantes (alumnos y profesores), que requiere soporte y emplea actividades de aprendizaje correctamente estructuradas.

Una plataforma basada simplemente en una página web podría proporcionar a los estudiantes los apuntes y una completa documentación adicional. Sin embargo, las plataformas y aplicaciones de *e-learning* disponibles hoy en dia ofrecen algo más, ofrecen una completa virtualización del proceso de enseñanza-aprendizaje, respondiendo a todos los requerimientos de profesores y alumnos. Algunos de estos requerimientos son la necesidad de restringir o controlar el acceso a un curso únicamente a estudiantes y profesores registrados en dicho curso, disponer de herramientas que permitan la comunicación de los miembros (mediante chats, e-mail, blogs, etc.), disponer de herramientas para la evaluación de los estudiantes, de un calendario que permita la planificación de todas las tareas y actividades asociadas al curso, y un largo etéctera.

Actualmente, el *e-learning* es una realidad en la mayor parte del sistema universitario español, aunque los objetivos y beneficios que cada institución académica busca en este tipo de enseñanza pueden ser algo diferentes. En este artículo describiremos la experiencia de la Universidad Politécnica de Cartagena (UPCT) en la implantación y utilización de dos plataformas de *e-learning*, una plataforma proprietaria llamada WebCT y una plataforma de libre distribución denominada Moodle. Por otro lado, el *e-learning* se ha convertido en un nuevo campo de investigación y desarrollo, no sólo de carácter pedagógico sino también técnico. El desarrollo y mejora de las aplicaciones asociadas al *e-learning* es un interesante y actual campo de trabajo. Este artículo presentará una contribución al mundo del *e-learning*. Describirá la potencialidad y detallará los aspectos técnicos más interesantes de Vicuc, un módulo que permite realizar audio/video conferencias desarrollado para la plataforma Moodle.

II. La experiencia de la UPCT en el e-Learning

La Universidad Politécnica de Cartagena (UPCT) actualmente está formada por unos 600 miembros de personal docente e investigado, distribuidos en 24 departementos y en 22 titulaciones. El número de estudiantes matriculados supera los 6000. Por tamaño y variedad de estudios, la UPCT representa un escenario adecuado para la implantación y estudio de una experiencia *e-learning*.

Como ya se ha comentado en la introducción, el *e-learning* es hoy en día una realidad en el sistema universitario español. La mayoría de las universidades españoles emplean de una forma u otra los beneficios de esta tecnología. Algunas universidades únicamente disponen de algúna herramienta basada en el uso de Internet como canal de distribución. Otras hacen

un uso más completo de las posibilidades del *e-learning*. El enfoque y los objetivos de cada institución son diferentes, abarcando un amplio abanico de posibilidades que van desde el uso del *e-learning* como complemento a la educación presencial hasta la implantación de una educación totalmente virtual (u *on-line*).

Tras analizar las diferentes plataformas de *e-learning* empleadas por las universidades españolas, podemos concluir que estas se pueden clasificar casi equitativamente en las siguientes opciones: universidades que han optado por desarrollar su propia plataforma de *e-learning*, universidades que emplean alguna solución propietaria (principalmente WebCT y en menor medida EduStance) y universidades que han optado por alguna plataforma de *e-learning* de libre distribución. No obstante, muchas de las universidades (incluyendo la UPCT) que inicialmente emplearon una solución de *e-learning* propietaria, están ahora apostando por un cambio hacia alguna de las otras dos opciones. En el caso de la Universidad Politécnica de Cartagena, la evolución ha sido de la plataforma propietaria WebCT a la plataforma de libre distribución Moodle.

II-A. WebCT

WebCT (*Web Course Tools*) [2] es un sistema de aprendizaje virtual propietario. Fue originariamente desarrollado en la Universidad de British Columbia (Canada) por un miembro de la facultad de informática, Murray W. Goldberg. Más tarde, en 2006, WebCT fue adquirido por la compañía Blackboard.

Esta plataforma presenta un entorno bastante fácil de usar para el desarrollo de cursos *on-line*, permitiendo la difinición de un escenario de enseñanza virtual completo. Los profesores pueden añadir a sus cursos creados en WebCT diferentes herramientas interactivas como blogs, e-mail, chats, contenidos en formato web, archivos PDF y otros formatos, etc. Por todas estas características, la UPCT decició en 2001 utilizar WebCT, bajo el nombre de AulaVirtual, para comenzar con su experiencia en el *e-learning*. Inicialmente, como experiencia piloto, sólo una asignatura fue ofrecida a los estudiantes mediante esta plataforma. En los siguientes años, el número de asignaturas que hacían uso de WebCT fue creciedo hasta alcanzar una cifra alrededor de 100 cursos en 2005.

Sin embargo, WebCT presenta algunas desventajas, siendo las más importantes las que se enuncian a continuación:

- WebCT requiere del pago de una liciencia por estudiante y por asignatura. Es decir, si un estudiante está matriculado de cinco asignaturas diferentes, serán necesarias cinco licencias.
- El sistema de administración de WebCT es confuso y algo complicado, ya que se basa en ficheros, en lugar de en bases de datos.
- Su interfaz gráfica es fija y predefinida. Por tanto, los formatos y los estilos no pueden adaptarse a la imagen corporativa de la universidad.
- WebCT es una plataforma propietaria. Por ello, la comunidad de usuarios de WebCT no pueden disfrutar de las ventajas de un entorno de trabajo de código abierto tradicional, donde existe la posibilidad de contribuir y hacer uso de nuevos desarrollos.

La experiencia de profesores y alumnos de la UPCT en el mundo del *e-learning* resultó muy satisfactoria, lo que se plasmó en el aumento año tras año del número de asignaturas (y de estudiantes) registrados en la plataforma de enseñanza virtual. Sin embargo, la potencialidad de WebCT quedó obsoleta con el paso del tiempo. Por un lado, la administración de la aplicación (creación de cursos, altas y bajas de alumnos, etc) se convirtió en una tarea dura y ardua para al administrador de WebCT. Este hecho es aún mas relevante considerando que la UPCT emplea un avanzado sistema de bases de datos para la administración del proceso de matriculación de alumnos. Por ello resultaba muy interesante que se pudiera implementar un mecanismo de sincronización entre el proceso de matriculación y la administración de los cursos en WebCT. Por otro lado, la UPCT deseaba extender su imagen corporativa a la plataforma de *e-learning* que empleaba, pero WebCT no permitía dar respuesta a este requerimiento.

Todos estas circunstancias hicieron que, en 2006, la UPCT considerara la migración a otra herramienta de enseñanza virtual. Tras analizar diferentes soluciones, la UPCT eligió la plataforma de libre distribución y código abierto Moodle como la mejor alternativa.

II-B. Moodle

Moodle [3][4] es una plataforma de *e-learning* gratuita y de código abierto basada en el modelo del constructivismo para el proceso de ensañanza-aprendizaje. Esta plataforma ha sido diseñada para ayudar a educadores a crear cursos *on-line* con un alto nivel de interacción.

Moodle fue creada por Martin Dougiamas, un administrador de WebCT de la Universidad Curtain de Australia. Dougiamas basó el diseño de esta plataforma en las ideas pedagógicas del constructivismo, haciendo énfasis en que los estudiantes (y no sólo los profesores) pueden contribuir a la experiencia docente de muchas maneras. Los principios del constructivismo afirman que el conocimiento se construye en un entorno de aprendizaje colaborativo y en la mente del estudiante, en lugar de ser únicamente transmitido desde libros o simples lecciones.

Desde su creación, la evolución y expansión de Moodle ha sido imparable. Hoy en día, hay más de 50.000 sitios registrados con más de 21 millones de usuarios registrados alrededor del mundo.

En términos tecnológicos, Moodle es una plataforma desarrollada bajo los términos de una licencia GNU GPL. La instalación es sencilla, requiriendo únicamente un servidor web que soporte PHP y la disponibilidad de una base de datos. Moodle tiene una capa de abstracción de bases de datos por lo que soporta los principales sistemas gestores de bases de datos (MySQL, PostgreSQL y Oracle son los más utilizados). Por lo tanto, Moodle puede ser instalado sobre Unix, Linux, FreeBSC, Windows, MacOS y cualquier otro Sistema Operativo que cumpla con las condiciones anteriores.

Las principales razones por las cuales la UPCT eligió Moodle como solución alternativa a WebCT son las siguientes:

- Moodle permite el desarrollo de actividades pedagógicas que respondan a los criterios establacidos por los planes de estudio de la UPCT, proporcionando herramientas específicas para la enseñanza.
- La curva de aprendizaje en el uso de la plataforma, tanto

para profesores como alumnos, es mínima e inversamente proporcional a sus beneficios educativos.
- Moodle requiere un proceso de instalación sencillo. Además, es escalable, lo que permite cualquier velocidad de crecimiento del campus virtual.
- Existe una gran comunidad internacional que ofrece acciones de soporte y mantenimiento (la comunidad de Moodle en España es también muy activa y numerosa). Por otro lado, los componentes involucrados en el software (lenguaje PHP, servidor Apache y bases de datos MySQL y Oracle) persentan un horizonte de suficiente estabilidad. Además, el conocimiento técnico de estas tecnologías permite que el mantenimiento de Moodle puede llevarse a cabo por el personal técnico de la propia universidad.
- Moodle es software libre. Por ello, desaparece el coste asociado al número de licencias requeridas, sin perder los beneficios asociados al propio dinamismo del desarrollo de la plataforma (continuas contribuciones, nuevos módulos, actualizaciones, etc.)

El apendice I muestra una tabla resumen de las principales características de Moodle. Tras analizar todas ellas y estudiar las herramientas que ofrece Moodle, podemos destacar una importante ausencia. No existe ningún módulo, herramienta o aplicación útil para realizar video-conferencias. Por esta razón, se consideró muy interesante el desarrollo de una herramienta de video-conferencia compatible con Moodle. La siguiente sección describirá en detalle las características de la herramienta desarrollada así como los principales aspectos técnicos asociados a su codificación.

III. VICUC: Descripción y funcionalidad

VICUC (**VI**deo **C**onferencia **U**niversidad Politécnica de **C**artagena) es un nuevo módulo (o actividad, en la argot de Moodle) que ofrece una interesante herramienta para la docencia: la audio/video conferencia. Como su nombre indica, esta actividad permitirá al profesor mantener una conferencia en tiempo real con sus estudiantes mediante la plataforma de *e-learning* Moodle. Además, la audio/video conferencia realizada podrá ser grabada, y gracias a ello podrá estar disponible para cualquier participante en la actividad para su posterior reproducción.

La herramienta de conferencia Vicuc permite dos modalidades: la realización de una audio-conferencia o la realización de una video-conferencia completa (es decir, con audio y video). En ambos casos Vicuc ofrece al profesor un apoyo extra. El profesor podrá emplear durante la conferencia una presentación basada en transparencias (tipo presentación power-point) para complementar la actividad.

También es importante destacar que esta herramienta de video-conferencia ofrece un canal de comunicación bidireccional (aunque asimétrico) entre los participantes: el profesor podrá hacer uso del audio y/o video para comunicarse con los estudiantes, y los estudiantes podrán usar un chat para comunicarse entre ellos o con el profesor.

La figura 1 muestra el aspecto de Vicuc una vez que una video-conferencia ha comenzado. Esta imagen corresponde al caso más general, es decir, una video-conferencia que hace uso de una presentación. La zona número 1 muestra la imagen del profesor durante la conferencia, capturada por una webcam. La zona número 2 ofrece una funcionalidad doble: la primera pestaña (Usuarios) lista el nombre de todos los usuarios participantes en la video-conferencia en ese momento y la segunda pestaña (Chat) ofrece un canal de comunicación entre los estudiantes y el profesor. Finalmente, la zona número 3 muestra la presentación powerpoint que el profesor está utilizando durante la video-conferencia para complementar su explicación o actividad.

Aunque durante la sesión, será el profesor quien controle el ritmo de la conferencia, avanzando las transparencias hacia delante o atras, los alumnos tienen cierta libertad de movimiento. Mientras el profesor está detenido en una transparencia, el alumno podrá consultar las transparencias anteriores si lo considera necesario. Sin embargo, cuando el profesor avance de transparencia, todos los alumnos se situarán inmediantamente en ese nuevo punto de la presentación. Además, el profesor tiene la posibilidad de detener y reanudar la conferencia en cualquier momento (mediante el botón de la esquina superior derecha).

III-A. Creación de una conferencia Vicuc

El bloque de Actividades que, por defecto ofrece Moodle, permite utilizar en el curso diferentes actividades tales como chats, cuestionarios, encuestas, foros, glosarios, recursos, wikis, etc. El nuevo módulo presentado en este artículo, Vicuc, aparecerá en dicho bloque como una nueva actividad disponible.

El profesor creará una instancia Vicuc seleccionando la actividad **Vicuc Video-Conferencia** en el menú desplegable de actividades. En ese momento tendrá que completar las siguientes opciones de configuración:
- Nombre: el nombre de la conferencia que los usurios del curso verán en el listado de actividades.
- Descripción: una pequeña descripción de los contenidos de la conferencia.
- Fecha de planificación: fecha en la que, a priori, comenzará la conferencia. Esta fecha no es de obligado complimiento, sino una fecha prevista a modo de información para todos los usuarios que deseen participar en la conferencia.
- Modo de presentación: tiene dos opciones: “Sólo para navegadores Internet Explorer” y “Para todos los navegadores”. El primer modo de presentación permite ver las diapositivas que forman la presentación únicamente a usuarios que posean el sistema operativo Windows y utilicen el navegador Internet Explorer (I.E). En este caso, Vicuc permitirá al profesor usar una presentación creada con el programa Microsoft PowerPoint. Para permitir mostrar el formato ppt, Vicuc utiliza un control *activex* que solo funciona en I.E. Este control ofrece al profesor todas las opciones disponibles en una presentación ppt normal. El segundo modo de presentación permite a cualquier usuario desde cualquier sistema operativo y utilizando cualquier navegador web participar en la conferencia. En este caso, una vez que el profesor ha indicado que fichero de presentación que desea exponer durante su conferencia, una aplicación de conversión se encargará automáticamente de convertir cada transparencia de la presentación en una imágenes en formato PNG. Aunque los aspectos tecnológicos se

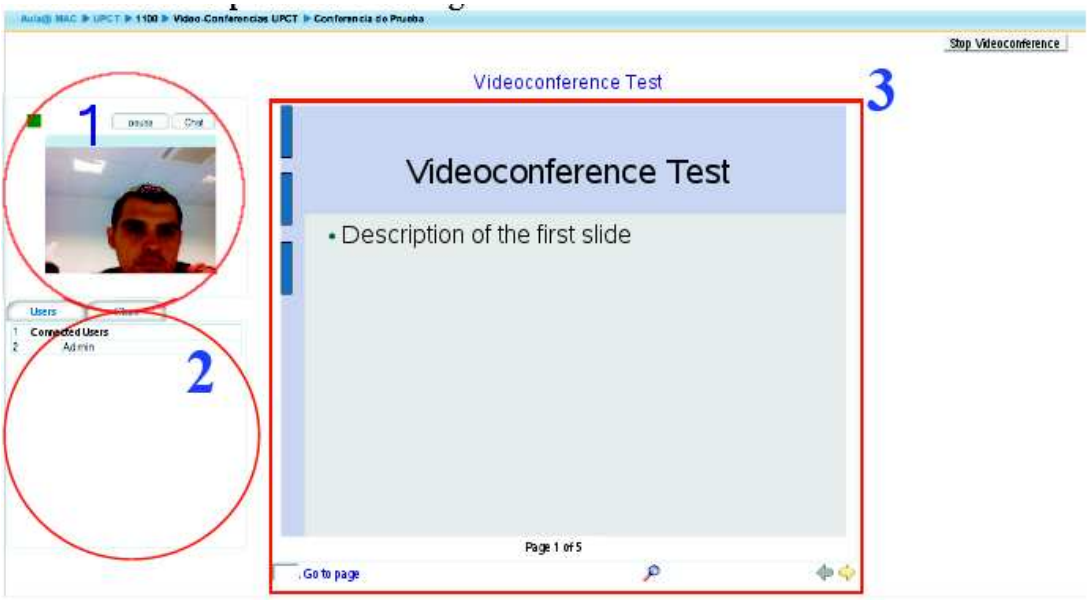


Figura 1. Interfaz gráfico de la aplicación Vicuc.

describirán mas tarde, se ha elegido el formato PNG por su buena relación tamaño/calidad.

- Archivo de presentación: esta opción permite al profesor, si lo desea, subir al servidor un fichero de presentación y enlazarlo con la conferencia que se está creando.
- Grabar conferencia: el profesor deberá especificar mediante un checkbox si desea que la conferencia sea grabada. Si es así, cuando finalice, los participantes podrán acceder a ella para visionarla de nuevo cuando deseen.
- Audio conferencia: el profesor deberá especificar mediante un checkbox si desea que la conferencia se realice únicamente con audio, para lo cual sólo será necesario un micrófono, o también con video, para lo cual será necesario una webcam.
- Visible: indicará si la conferencia está visible para los alumnos, o si por el momento el profesor prefiere mantenerla oculta.

III-B. Una actividad Vicuc en ejecución

Una vez creada la conferencia, y si ésta es visible, los alumnos verán una nueva tarea planificada en el calendario general del curso, así como una nueva actividad en la página principal del curso.

Figura 2. Icono de una video-conferencia Vicuc.

Cuando el profesor desee comenzar la conferencia, irá a la página principal del curso, donde encontrará el icono de la nueva actividad Vicuc Video-Conferencia (ver figura 2). Al hacer click sobre esta actividad, se abrirá una página que mostrará al profesor un enlace para comenzar la conferencia. Cuando el profesor pulse sobre este enlace, se iniciará la conferencia, y entonces, los alumnos que lo deseen podrán unirse a ella.

Par unirse a una conferencia, un estudiante deberá cliquear sobre el enlace asociado a la actividad disponible en la página principal del curso. Este enlace abrirá una página en la cual el alumno puede ver el nombre de la conferencia así como su descripción y además podrá ver el listado de usuarios que se encuentran en línea en la conferencia en ese momento. Finalmente, al estudiante deberá pulsar en el enlace con título "Pulse para unirse a la conferencia" de dicha página.

III-C. El interfaz Vicuc

Como se ha descrito anteriormente, una conferencia Vicuc puede realizarse sólo con audio, con audio y video y, adicionalmente, puede emplear una presentación para complementar la charla del profesor. Por ello, la interfaz gráfica que verán el profesor y los alumnos variará dependiendo de las características de la conferencia.

Video-conferencia con archivo de presentación: la interfaz del profesor correspondiente a este caso se muestra en la figura 1. Como se describió anteriormente, la interfaz gráfica se divide en tres áreas: la zona de imagen, la zona de chat y usuarios, y la zona de la presentación. Esta última zona ofrece al profesor las fechas de avance y retroceso que le permitirán desplazarse por las transparencias que componen la presentación. La interfaz que verán los alumnos es muy similar a la del profesor. La única diferencia es la ausencia del botón que parar/continuar que permite al profesor detener y renaudar la conferencia.

Video-conferencia sin archivo de presentación: en este caso, al no emplearse una presentación durante la conferencia, la interfaz elimina la zona de las diapositivas. Al eliminar el visor de presentación de la interfaz, ésta se reestructura mostrando a la izquierda la zona de imagen, y a la derecha (en un tamaño mayor que en el caso anterior) la zona de chat y usuarios.

Audio-conferencia: esta opción es válida independientemente de la inclusión o no de una presentación. En este caso, al elegir la opción sólo audio, se desactiva el vídeo y aparece en su lugar el siguiente mensaje: "Conferencia sin video. Sólo AUDIO".

III-D. Reproducción de una conferencia grabada

Una vez la conferencia Vicuc ha finalizado, tanto profesor como alumnos podrán acceder a un historio de la actividad, en el que por defecto se mostrarán los mensajes que se generaron en el chat y el listado de participantes en la conferencia. Si el profesor optó por grabar la conferencia, la aplicación almacenará también el audio/video y los cambios entre diapositivas.

Tanto los alumnos como el profesor podrán conocer si la conferencia se está grabando gracias a un icono parpadeante que aparece en la esquina superior-izquierda de la interfaz Vicuc. Una vez finalizada la conferencia, tanto el profesor como los alumnos podrán visualizar una reproducción en diferido. En este caso, la interfaz Vicuc es similar a la de una conferencia en tiempo real. La única diferencia es la presencia de los botones de Play/Pause, así como una barra de desplazamiento que mostrará el punto en el que se encuentra la reproducción. Además no se ofrece la posibilidad de desplazamiento por las diapositivas, ya que el control lo tiene la propia reproducción. El contenido del chat y los usuarios participantes también se mostrarán.

IV. Architectura Tecnológica de Vicuc

Vicuc es un módudo de audio/video conferencia desarrollado en flash y php. Además, utiliza la tecnología Java (junto con la ayuda de algunas APIs de OpenOffice) para la transformación de las presentaciones en formato PowerPoint a imágenes.

La elección de la tecnología flash se basa en su alcance. El *plug-in* de flash se encuentra instalado por defecto en la mayoría de los navegadores, y si no es así, la mayoría de ellos ofrecen una herramienta que permite la descarga e instalación del *plug-in* de una manera fácil y sencilla.

El módulo Vicuc para Moodle requiere de la creación de cuatro nuevas tablas en la base de datos de Moodle:

- La tabla principal del módulo Vicuc (m_vicuc). Esta tabla es la encargada de guardar cada conferencia creada y sus características (nombre, introducción, si será grabada o no, ...)
- La tabla de usuarios Vicuc (m_vicuc_users). Esta tabla almacena el histórico de los usuarios que han accedido a cada conferencia Vicuc.
- La tabla de mensajes Vicuc (m_vicuc_messages). Esta tabla es la encargada de almacenar todos los mensajes de chat que han sido escritos durante cada conferencia Vicuc.
- La tabla de eventos Vicuc (m_vicuc_record). Esta tabla almacena los eventos que se producen durante una conferencia. Únicamente se almacenará información si la conferencia está siendo grabada.

La figura 3 muestra el diagrama entidad-relación de las cuatro tablas vicuc y la tabla de usuario de Moodle.

El módulo Vicuc también requiere actualizar un fichero de configuración con dos parámetros: la dirección IP y el número de puerto del servidor flash. Este servidor es el encargado de servir el streaming compartido de audio y video durante la conferencia, así como de mantener en sesión los objetos compartidos necesarios para controlar la conferencia. En nuestro caso, el módulo Vicuc empleará un servidor flash de libre distribución denominado Red5. El acceso a los parámetros de configuración se realizará mediante el menú de administración de Moodle.

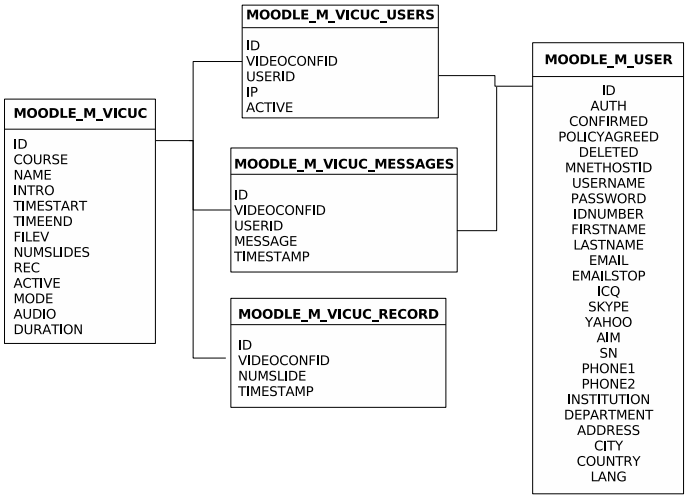


Figura 3. Diagrama entidad-relación.

IV-A. Servidor de Streaming Flash : Red5

Red5 [5] es un servidor flash de libre distribución programado en Java que permite el streaming de audio y video (formatos flv y mp3). Red5 posee las cualidades más importantes del servidor Flash Media Server (FMS) de Adobe, uno de los servidores de streaming más utilizados. Aunque FMS es un servidor más robusto que Red5, su coste es demasiado elevado para desarrollar aplicaciones pequeñas. Tanto FMS como Red5 emplean el protocolo propietario RTMP (*Real Time Messaging Protocol*) desarrollado por Adobe System para realizar el streaming de audio, video y datos sobre Internet, entre el reproductor de flash y el servidor.

IV-B. Grabación y reproduccion de una conferencia

Como se indicó en la sección anterior, el profesor puede optar por grabar la conferencia. En este caso, el servidor flash es el encargado de recoger el audio/video de la webcam del profesor, y de grabar toda esta información en un archivo situado en un directorio del servidor flash Red5 denominado oflaDemo. El fichero tendrá extensión .flv (formato de video propietario de Flash) y estará identificado por el identificador de la conferencia que se está grabando.

Además de este fichero, todos los demás eventos que se produzcan durante la conferencia, como la entrada y salida de participantes, los mensajes de chat e incluso el cambio de diapositivas por parte del profesor serán almacenados en una base de datos con su marca temporal correspondiente para permitir la posterior reproducción.

Para realizar la reproducción, en primer lugar el reproductor flash cargará un archivo XML con todos los eventos. Este archivo XML será generado a partir de la información almacenada en la base de datos cada vez que se pulse el enlace que

permite visualizar la reproducción. La localización del archivo XML es el servidor de aplicaciones, en el directorio de datos de Moodle, dentro del directorio del curso y a su vez en el directorio donde se almacena la presentación empleada en la conferencia. Una vez el archivo XML se ha creado y se inicie la página web del reproductor, el reproductor flash cargará el archivo añadiendo eventos (*cuePoints*) al vídeo antes de que comience la reproducción. Estos *cuePoints* permitirán al reproductor identificar cuándo va a suceder un evento, y con ello, justo en el instante de tiempo adecuado, llamar mediante ActionScript [6] a un método javascript de la página para reflejar el nuevo evento.

IV-C. Conversión de una presentación powerpoint

Como se ha descrito en la sección III, cuando un profesor crea una conferencia Vicuc, puede seleccionar si la presentación que va a emplear asociada a la conferencia es adecuada "Para todos los navegadores" o si es "Sólo adecuada para Internet Explorer". En el primer caso, el fichero de presentación en formato powerpoint (ppt) debe ser transformado en un nuevo fichero de imágenes (una imagen por cada transparencia). Para realizar esta conversión, el módulo Vicuc hace uso de una API proporcionada por el software de libre distribución OpenOffice denominada UNO [7]. Por ello es necesario que el programa OpenOffice esté configurado en el lado del servidor y ejecutándose en modo oculto en el momento en el que se desea hacer la transformación. Vicuc se comunicará con OpenOffice mediante el protocolo remoto URP (UNO Remote Protocol). La aplicación que realiza la conversión ha sido desarrollada en Java, siendo necesario incluir las librerías para Java de UNO.

IV-D. Xajax

Ajax es una tecnología que utiliza a su vez otra combinación de tecnologías, como XML y Javascript para realizar peticiones de contenido o de computación a un servidor sin tener que recargar la página en la que está el usuario. Es una tecnología que permite una nueva gama de aplicaciones web interactivas, mucho más ricas y rápidas, dado que no es necesario recargar todo el contenido de una página para realizar peticiones al servidor.

Xajax [8] es una biblioteca de código abierto de PHP capaz de generar aplicaciones Web con tecnología Ajax. Con Xajax podemos fácilmente ejecutar funciones PHP, que se ejecutan en el servidor, cuando el usuario realiza acciones en la página. Luego, los resultados de esas funciones PHP se producen en la misma página, sin que se tenga que recargarse. Xajax es un producto Open Source gratuito y compatible con los navegadores más comunes, como Firefox, u otros navegadores basados en Mozilla, Internet Explorer, Opera, etc.

En la aplicación Vicuc, Xajax está presente durante la conferencia tanto en la parte del profesor como de los alumnos. Las funcionalidades del módulo Vicuc implementada mediante Xajax son: el almacenamiento de los mensajes del chat en la base de datos en tiempo real y la desconexión de alumnos y profesor a la conferencia.

V. CONCLUSIONES

El *e-learning* es una realidad en el sistema educativo actual y, en particular, en el sistema universitario. Este artículo ofrece una visión de la evolución en la implantación y uso del *e-learning* en un escenario universitario real, mostrando los problemas y limitaciones que pueden surgir. Nuestra experiencia demuestra que es posible ofrecer un sistema de *e-learning* de calidad sin necesidad de utilizar un sistema propietario. Aun mas, la utilización de una plataforma de libre distribución hace posible la mejora de su funcionalidad, mediante el desarrollo de nuevas herramientas como por ejemplo el desarrollo del módulo de audio/video conferencia Vicuc para la plataforma Moodle. Vicuc se ha desarrollado utilizando únicamente tecnologías actuales y extensamente conocidas, tales como Flash, PHP y Java. El nuevo módulo permite a profesores crear audio/video conferencias, con la posibilidad de apoyarse en un conjunto de transparencias tradicionales como material complementario, y además ofreciendo un canal de comunicación bidireccional entre todos los participantes.

AGRADECIMIENTOS

Los autores desearían expresar su agradecimiento a los miembros del Servicio de Informática de la Universidad Politécnica de Cartagena, especialmente a D. Antonio Máximo González-Adán por su trabajo y sincera cooperacion, comentarios y sugerencias. Este trabajo ha sido subvencionado por el proyecto nacional TEC2007-67966-C03-01/TCM (CONPARTE-1) y ha sido desarrollado en el marco del "Programa de Ayudas a Grupos de Excelencia de la Región de Murcia, de la Fundación Séneca, Agencia de Ciencia y Tecnología de la RM (Plan Regional de Ciencia y Tecnología 2007/2010)".

REFERENCIAS

[1] Kahiigi, E.K, et al, "Exploring the e-Learning State of Art", *The Electronic Journal of e-Learning*, 2008, Volume 6, Issue 1, pp. 77-88.
[2] WebCT Home Page. http://www.webct.com/webct
[3] Moodle Home Page. http://www.moodle.org
[4] UPCT E-learning Home Page. http://moodle.upct.es
[5] Red5 Open Source Flash Server Home Page. http://osflash.org/red5
[6] ActionScript Technology Home Page. http://www.adobe.com/devnet/actionscript/
[7] OpenOffice UNO API Home Page. http://api.openoffice.org/docs/common/ref/ com/sun/star/module-ix.html
[8] Xajax Home Page. http://xajaxproject.org

APENDICE I.

Nombre LMS	Moodle
Equipo de desarrollo	En 1999 Martin Dougiamas (Australia) inició el desarrollo de Moodle. Actualmente colaboran con él alrededor de 100 personas entre desarrolladores, traductores, beta-testers,...
Servicio de soporte	Desde 2003 en http://www.moodle.com los Moodle Partners ofrecen una amplia oferta de servicios comerciales para usuarios, entre los cuales está el hosting completo de Moodle, contratos de soporte remoto, desarrollos a medida y consultoría.
Evaluated Version	1.9.4
Tecnología utilizada	PHP
S.O. Soportados	Unix, Linux, Windows, MacOS X, Netware y cualquier otro que soporte PHP
Requisitos de bbdd/software	Apache, PHP 5.x o posterior, y abstracción de bases de datos (p.e. MySQL, Oracle).
Licencia	GPL
Estándares de e-learning soportados	SCORM 1.2 and IMS-QTI
Lenguas disponibles	Más de 60 lenguas
Características principales	La principal ventaja de Moodle, aparte de su fundamento en la pedagogía del construc-tivismo social es su gran y continuamente creciente comunidad de usuarios que le dan al sistema una enorme vitalidad
Posibilidad de gestionar los contenidos de los cursos	Moodle permite crear y gestionar contenidos de cursos y tests
Aspectos positivos	- El sistema es muy intuitivo y fácil de usar - Está traducido a más de 60 lenguas - Se apoya en una gran comunidad de usuarios y desarrolladores - Personalización de la apariencia mediante un sistema de plantillas - Repositorio de módulos desarrollados - Revisión contínua de versiones - Multi-autenticación de usuarios - Soporte para SCORM 1.2 e IMS Content Packaging
Aspectos a mejorar	- Está en marcha el proyecto de mejora de la documentación Moodle Documentation Project basado en wiki - Problemas con el editor HTML incorporado - No posee herramientas de audio/video conferencia - Mejorar la documentación para desarrolladores
Principales instalaciones en el mundo	Miles de instalaciones listadas http:www.moodle.org/sites
Enlace a la versión demo	http://www.moodle.org

Mejoras en OPNET para el dimensionamiento del tráfico agregado en VoIP

Juan Jiménez, Antonio Estepa, Germán Madinabeitia, Rafael Estepa
Área de Telemática, Escuela Superior de Ingenieros
Universidad de Sevilla
C/ Camino de los descubrimientos s/n., 41092 Sevilla (Spain).
E-mail:{juanjimenez, aestepa, rafa, german}@trajano.us.es

***Resumen*—Algunos codecs usados en VoIP como el G.729AB y el AMR permiten la generación de ruido de confort (CNG) durante los periodos de inactividad vocal. Esta característica altera el modelo de fuente ON-OFF clásico utilizado para modelar fuentes VBR (i.e. tráfico VoIP con supresión de silencios).**

En este artículo, proponemos un nuevo modelo de fuente de VoIP con capacidad CNG. Además implementamos este modelo en el simulador OPNET, permitiendo el uso de codecs VBR y VBR+CNG. Los resultados se validan usando trazas de tráfico real y muestran una mejora importante con respecto al modelo ON-OFF tradicional, haciendo más precisa la caracterización de tráfico de una fuente y el dimensionamiento de tráfico VoIP agregado.

***Palabras Clave*—VoIP, modelos de multiplexión ON-OFF, dimensionamiento VoIP.**

I. Introducción

La voz sobre IP (VoIP) requiere unas garantías mínimas de servicio que la estructura *best-effort* de las redes IP de hoy en día no puede ofrecer. Para satisfacer esas garantías es necesario el uso de técnicas adicionales como MPLS, DiffServ o IntServ que permiten cumplir las exigencias necesarias de aplicaciones en tiempo real como las de VoIP. Independientemente de qué técnica se use, una caracterización precisa del tráfico generado por los terminales VoIP (es decir, codificación y esquema de empaquetado) es clave en los mecanismos de QoS aplicados.

G.729AB, G.723.1 o *Adaptative Multi Rate* (AMR) son algunos de los codecs ampliamente usados hoy en día en las aplicaciones de VoIP[1]. Estos codecs tienen la capacidad de detectar la inactividad vocal para evitar enviar datos durante los periodos de silencio (característica de supresión de silencios), lo que permite reducir el ancho de banda utilizado en un 50% aproximadamente. Además de la supresión de silencios, estos codecs también incluyen nuevas características como generación de ruido de confort (CNG) que mejora la naturalidad de las conversaciones. Esta característica hace uso de tramas de descripción de silencio (SID), que transportan información del ruido de fondo, y que se envían durante los periodos de inactividad para mejorar la percepción del oyente. Las tramas SID se envían al inicio de un periodo de silencio o cuando las características del ruido cambian notablemente [1].

Los modelos actuales usados para caracterizar el tráfico VoIP están basados en modelos ON-OFF que no tienen en cuenta la generación de tramas SID. Esto puede deberse al hecho de que la mayoría de los trabajos existentes son anteriores a la estandarización de estos codecs o al hecho de que el tamaño de las tramas SID es pequeño frente a las tramas de voz, con lo que el aumento que provocan en la tasa media de salida del codec es sólo de un 1% a un 3% [2]. Sin embargo, en los paquetes que se envían a la red, las cabeceras pueden ser varias veces más grandes que las tramas de voz. Por tanto, las tramas SID pueden tener un impacto serio en la tasa media del tráfico enviado a la red y provocar errores en la caracterización tradicional de tráfico de voz en redes IP [2], [3].

El estudio sobre el comportamiento de tráfico agregado de N fuentes de voz en un multiplexor estadístico es de especial interés para técnicas de QoS como DiffServ. Los recursos asignados al tráfico de VoIP (i.e. ancho de banda) serán claves para poder asegurar que se satisfacen nuestras exigencias en el nodo multiplexor. Por tanto, el error producido al ignorar las tramas SID puede causar una violación del perfil de tráfico declarado en un acuerdo de nivel de servicio (SLA).

Existen dos alternativas tradicionales para el estudio del dimensionamiento del tráfico agregado de VoIP: la elección de modelos analíticos o la simulación. Los modelos analíticos ofrecen una mayor generalidad y son plausibles en escenarios simples, mientras que la simulación es usada cada vez con mayor frecuencia debido a la alta precisión de los resultados en escenarios complejos. En nuestro trabajo previo [1] dimos una solución analítica al problema del dimensionamiento cuando los codecs generan tramas SID. No obstante no existen simuladores que incluyan este patrón de tráfico en sus modelos y por lo tanto válidos para dimensionar cualquier tipo de tráfico de VoIP.

OPNET es un simulador de redes muy extendido, válido para el análisis de dimensionamiento de aplicaciones de VoIP. El inconveniente es que el modelo que usa para generar el tráfico de voz se basa en el tradicional ON-OFF. Este patrón describe fielmente el comportamiento de codecs que generan tasas de bits constantes (es decir, codecs CBR) y también el de codecs que generan tasas variables (es decir, codecs VBR), pero no, el de aquellos que poseen características CNG (es decir, generación de tramas SID).

Por lo tanto, nuestro objetivo es implementar ese nuevo modelo en OPNET, extendiendo el modelo actual y consiguiendo describir un patrón de generación de tráfico más general que permita representar cualquier tipo de codec usado en VoIP. Además de la ausencia de tramas SID, el modelo de generación de tráfico de OPNET contiene otras imprecisiones

[1]El codec iLBC también es usado ampliamente en Internet debido a su buen mecanismo de recuperación ante pérdidas. Sin embargo, presenta una tasa más alta que la mayoría de los codecs VBR. Esto lo hace menos adecuado para entornos de tráfico de voz agregado.

que también son corregidas. Los resultados son validados con trazas de tráfico real, en primer lugar para el caso de una fuente de voz y posteriormente también son aplicados al dimensionamiento de tráfico agregado. En ambos casos, el modelo propuesto ofrece resultados más cercanos a las trazas que el modelo original ON-OFF.

El resto de este artículo está organizado de la siguiente manera. Primero, introducimos brevemente las bases de la caracterización de tráfico VoIP y nuestro modelo propuesto para la generación de tráfico VoIP: el modelo ON-SID. Luego, se describe el modelo de generación de tráfico de VoIP en OPNET, se enumeran los cambios necesarios para crear el modelo ON-SID propuesto, se añaden nuevos codecs con capacidad CNG y se mejora el modelo ON-OFF original. Por último presentamos los resultados para una fuente de voz y para el tráfico agregado. Estos últimos pueden aplicarse al dimensionamiento en un multiplexor estadístico.

II. CARACTERIZACIÓN DEL TRÁFICO EN VOIP

Los codecs de voz actuales pueden clasificarse como CBR o VBR en función del patrón de tráfico que generan. El tráfico generado por estos codecs ha sido modelado tradicionalmente como un proceso de nacimiento y muerte con dos estados (ON y OFF) llamado fuente ON-OFF. Se asume que el tiempo de permanencia en cada estado está distribuido exponencialmente[2] con medias 1/α (estado ON) y 1/ β (estado OFF) [4]. Las fuentes ON-OFF generan R_{on} bits/s. en el estado ON y ningún tráfico en el estado OFF. Por tanto, el régimen binario medio $\overline{R}$ de este modelo es:

$$\overline{R} = \frac{\alpha^{-1}}{\alpha^{-1} + \beta^{-1}} \cdot R_{on} = \rho \cdot R_{on} \quad (1)$$

donde ρ es el factor de actividad de la fuente. Valores típicos de α y β para el codec G.729 pueden encontrarse en [1], [2]. Debemos notar que haciendo ρ igual a uno el modelo ON-OFF se hace válido también para los codec CBR (p.ej. G.711). Además de la supresión de silencios, codecs de audio modernos como el G.729AB o el AMR pueden mejorar la calidad de las conversaciones reproduciendo el ruido de fondo de éstas. Esta característica se consigue con el uso de un tipo especial de trama llamada SID, generada en el lado del hablante y que describe las características principales del ruido de fondo. Las tramas SID son generadas durante los periodos de inactividad vocal y su esquema de codificación difiere del de las tramas de voz (tramas ACT). Las tramas SID son generadas por el algoritmo de transmisión discontinua del codec (DTX) como respuesta a cambios en la energía del ruido de fondo y en función de reglas específicas que dependen de la implementación de cada codec [5]. El patrón de generación de tramas SID que depende del algoritmo DTX de cada codec puede resumirse como sigue:

- AMR: en este codec las tramas SID son enviadas regularmente al principio de cada periodo de silencio, dos tiempos de trama más tarde y después, cada 7 tiempos de trama. Este es el patrón que se sigue durante la duración de cada periodo OFF.
- G.729AB: Una trama SID se manda al principio de cada periodo de silencio. Los envíos posteriores son realizados como respuesta a cambios en la energía espectral del ruido de fondo. Por tanto el tiempo de envío de estas tramas puede considerase aleatorio.

Para capturar el efecto de las tramas SID en el patrón de tráfico generado por una fuente de voz, en anteriores trabajos [1] propusimos la sustitución del modelo ON-OFF por un modelo que pueda generar tráfico SID durante los estados OFF (modelo ON-SID). La Fig. 1 muestra nuestro modelo ON-SID donde el régimen binario medio durante el estado OFF es R_{off}, a diferencia del modelo ON-OFF tradicional donde era cero. Por tanto el régimen binario medio de nuestro modelo ON-SID queda:

$$\overline{R} = \rho \cdot R_{on} + (1 - \rho) \cdot R_{off} \quad (2)$$

Aplicando el teorema elemental de nacimiento y muerte, R_{off} puede expresarse como:

$$R_{off} = \frac{S_{SID}}{T \cdot E[X]} \quad (3)$$

donde *T* es el tiempo de trama del codec, *X* es una variable aleatoria discreta que modela el tiempo entre llegadas de tramas SID (en número de tiempos de trama *T*) y S_{SID} es el tamaño de la trama SID. La distribución experimental de *X* para distintos codec en un entorno típico de oficina puede encontrarse en [2], donde *E[X]* toma valores de 7,78 y 7,47 para los codecs G.729AB y AMR respectivamente.

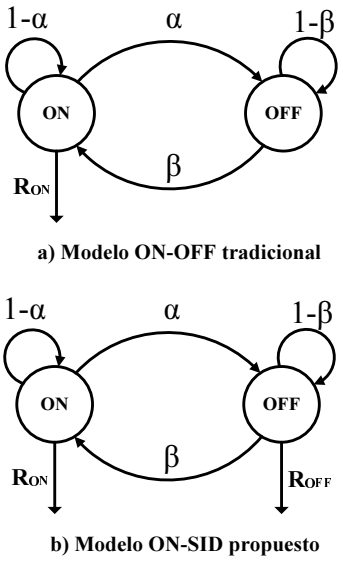


Fig. 1. Diagrama de estados de los modelos ON-OFF y ON-SID

Si definimos γ como el radio R_{off}/R_{on}, entonces $\overline{R}$ puede expresarse como:

$$\overline{R} = R_{on} \cdot [\rho + (1 - \rho) \cdot \gamma] \quad (4)$$

Por lo tanto, una fuente de VoIP queda definida por la cuádrupla $(\alpha, \beta, R_{on}, \gamma)$. Típicamente los valores de γ van desde 0,05 para el AMR hasta 0,1 para el codec G.729 [1], aunque el valor depende del número de tramas de voz transportadas en cada paquete IP (configurable en los terminales

[2]Aunque se ha comprobado que distribuciones como la Weibull ofrecen resultados más precisos, la distribución exponencial es muy utilizada a causa de su mejor tratabilidad matemáticamente.

VoIP). Hay que notar que si $\gamma = 0$ modelamos el comportamiento de un codec VBR, mientras que para $\gamma = 0$ y $\rho = 1$ modelamos un codec CBR.

III. Dimensionamiento de tráfico de voz agregado

El problema del dimensionamiento consiste en estimar el ancho de banda que debe reservarse para un tráfico agregado de N flujos de VoIP si pretendemos proporcionar unas prestaciones mínimas en un nodo multiplexor (p.ej. retardo máximo y porcentaje de pérdidas).

Una práctica común consiste en reservar el ancho de banda equivalente a la suma de las tasas de pico de los flujos (p.ej. $C = N \cdot R_{on}$). Esta aproximación asume el uso de fuentes CBR que son idóneas para LANs de alta velocidad [6], pero provocan un uso ineficiente de los recursos cuando se multiplexan fuentes VBR. En entornos WAN, los codecs con supresión de silencios (modelados típicamente como fuentes ON-OFF) pueden usarse para ahorrar ancho de banda, en estos escenarios es donde se hacen más patentes los beneficios de un buen dimensionamiento. Los métodos de dimensionamiento pueden estar basados en modelos analíticos o en el uso de simuladores.

A. Basado en modelos analíticos

Para obtener modelos de dimensionamiento para el tráfico VBR necesitamos hacer uso de modelos analíticos de multiplexión ON-OFF [4], [7], que pueden pertenecer a una de las siguientes familias: modelo de fluidos o modelos MMPP (Markov Modulated Poisson Processes) [4], [8]. De las soluciones de estos modelos se obtiene una probabilidad de desbordamiento de un buffer que se vacía a una velocidad constante en un nodo multiplexor cargado con un tráfico de N fuentes ON-OFF [9], [10], [11].

Después de desarrollar nuestro nuevo patrón de tráfico para una fuente (modelo ON-SID) parece lógico preguntarse que impacto tendrán las tramas SID en el tráfico agregado. Nuestro trabajo anterior [12] extiende los modelos MMPP y de fluidos incluyendo en éstos el patrón de generación de las fuentes ON-SID.

Los modelos analíticos proporcionan una estimación del nivel de QoS en un nodo multiplexor (p.ej. porcentaje de pérdidas) de forma directa, pero para los propósitos del dimensionamiento estamos interesados en resolver el problema desde un punto de vista opuesto. Nos interesa conocer el ancho de banda necesario para obtener un determinado nivel de QoS. Por tanto, necesitamos invertir las expresiones de pérdidas mencionadas anteriormente. Proponemos hacer esto usando cualquier método de calculo de raíces (p.ej. el método de la bisectriz) para iterar sobre la variable ancho de banda y encontrar el valor exacto que garantiza una cierta probabilidad de pérdidas en un nodo multiplexor (i.e un *router* de acceso). El retardo en el nodo puede modelarse con una cota superior que sería el tiempo que tarda en vaciarse el buffer lleno.

Consecuentemente, necesitamos averiguar la probabilidad de pérdidas en cada iteración ya sea usando expresiones analíticas o a través de las simulaciones. Un pseudocódigo detallado usado para invertir la expresión de pérdidas de forma iterativa puede encontrarse en [12].

B. Basado en el uso de simulador

Este método hace uso de simuladores que intentan reproducir el comportamiento real de una red de datos. Permite el análisis de escenarios complejos de forma rápida y sencilla y es complementario al uso de métodos analíticos de modelado. Los dos simuladores de red más significativos en la actualidad son NS-2 y OPNET. Ambos implementan fuentes de tráfico ON-OFF con los que se puede simular el comportamiento de codecs CBR y VBR. No obstante ninguno implementa la característica de generación de tramas SID.

Por ello, el resto de este artículo se centra en describir las modificaciones realizadas al simulador OPNET a fin de mejorarlo en dos sentidos: la corrección de fallos detectados que producen ligeras desviaciones en la estimación de la tasa media de una fuente de voz, y la modificación del modelo de fuente de voz para que incluya el patrón de tráfico de aquellos codecs que generan tramas SID. La decisión de OPNET sobre NS-2 se produce por simplicidad de uso y porque es igualmente gratuito para el ámbito universitario.

IV. Modelo de generación de tráfico en OPNET

OPNET usa un modelo de generación de tráfico basado en el modelo ON-OFF tradicional. El proceso encargado de implementarlo, *gna_voice_calling_mgr* presenta el diagrama de estados mostrado en la Fig. 2. Los estados principales del proceso son:

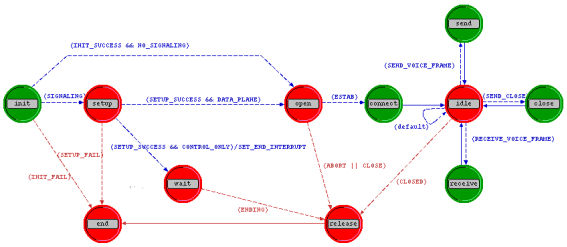


Fig. 2. Modelo de proceso *gna_voice_calling_mgr*

- *Init*: estado inicial donde se obtiene información sobre los atributos de la aplicación de voz tales como protocolo de transporte, puerto remoto o equipo de destino.
- *Setup*: asigna el tipo de señalización utilizada que puede ser SIP o ninguna.
- *Open*: se estable la sesión RTP.
- *Connect*: Tras iniciar la sesión RTP se establece el estado inicial de la fuente a OFF.
- *Idle*: estado que sirve como transición a los estados *send* y *received*.
- *Received*: se activa cada vez que se recibe un paquete y se usa para destruirlos.
- *Close*: cuando la aplicación finaliza, aquí se cierra la sesión RTP.
- *Release*: se ejecuta después de *close* para finalizar la sesión de señalización.
- *End*: una vez finalizada la aplicación, este estado libera la memoria y destruye el proceso.
- *Send*: Este estado juega un papel principal en nuestro trabajo, ya que controla cómo y cuándo se envían las tramas, modelando la fuente ON-OFF. Para la generación de tramas, planifica interrupciones cada vez que una

debe ser enviada. El estado es ejecutado después de cada interrupción. Al principio de la ejecución del estado, se comprueba si el estado actual es ON u OFF. Durante los periodos ON, se envía una trama de voz cada T (tiempo de trama) segundos, durante los periodos OFF no se envía nada. Un diagrama de flujo detallado se puede ver en la Fig. 3.

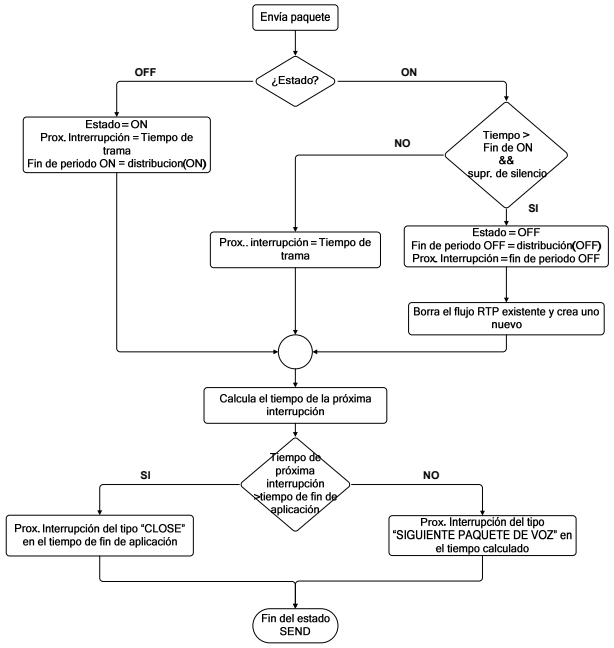


Fig. 3. Estado *send* en el modelo de fuente ON-OFF de OPNET

Como se dijo anteriormente, el modelo ON-OFF no tiene en cuenta la generación de tramas SID y en consecuencia, no puede modelar el patrón de tráfico de aquellos codecs que incorporan CNG.

A. Inclusión del modelo ON-SID en OPNET

Para poder implementar nuestro modelo ON-SID y los correspondientes codecs con CNG, necesitamos realizar las siguientes tareas:

1) Modificar la estructura de datos que modela las características de los codecs para extenderla con nuevos parámetros (p.ej. tamaño de las tramas SID).
2) Crear los codecs G.729AB y AMR que se ajustan a nuestro patrón ON-SID.
3) Modificar el modelo de proceso *gna_voice_calling_mgr* para conseguir un comportamiento acorde con nuestro modelo ON-SID.

A continuación se muestran en detalle las acciones realizas en cada una de las tareas.

1) Modificación de estructuras de datos: Para añadir a las estructuras parámetros relativos a SID modificamos el fichero *oms_dat_def_ds_desfs.h* añadiendo un campo llamado *sid_capable* que inidica si el codec tiene capacidad de enviar tramas SID o no. La estructura queda como sigue:

```
typedef struct
 {
  char*    name; /*Nombre del codec*/
  double   frame_size; /*Tamaño de trama de voz*/
  double   lookahead_size; /*Tiempo extra para prediccion*/
  double   dsp_proc_ratio; /* Tiempo de analisis*/
  double   speech_rate; /*Tasa durante actividad vocal*/
  double   silence_rate; /*Tasa durante silencios*/
  Boolean  sid_capable; /*Indica si el códec envía SID*/
 } GnaT_Voice_Encoder_Params;
```

También necesitamos definir las características de la aplicación de voz que se establecen en el fichero de cabecera *gna_mgr.h* que incluye la distribución del tiempo entre llegadas de tramas SID. Para ello añadimos dos campos a la estructura original: una cadena que almacena el nombre de la distribución y su manejador asociado.

```
typedef struct
{
    .
    .
/*nombre de la distribución de tiempo entre tramas SID*/
char* sid_interarrv_time_dist_str;

/* manejador de la distribucion */
OmsT_Dist_Handle sid_interarrv_time_dist_handle;
} GnaT_Voice_Desc;
```

2) Creación de los codecs G.729AB y AMR: Creamos dos nuevos codecs teniendo en cuenta sus parámetros de funcionamiento e incluimos dentro de sus atributos la capacidad de generación de tramas SID (ver Fig. 4). El codec AMR tiene 11 tasas distintas de funcionamiento de las que nosotros usaremos tan sólo la más alta: 12,2 Kbps y las más baja: 4,75 Kbps.

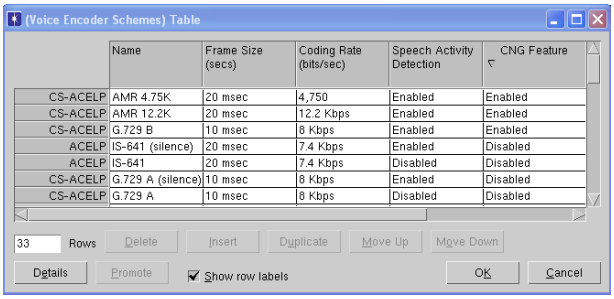


Fig. 4. Adición de la característica CNG a los codecs

3) Modificación del modelo gna_voice_calling_mgr *para seguir el patrón de tráfico ON-SID:* Los cambios pueden resumirse como sigue:

- Reemplazo de los ficheros de cabecera originales por los modificados.
- Modificación de aquellas funciones definidas dentro de *Function Block* que afecten al modelo.
- Modificación del estado *send* para adaptarlo al nuevo modelo como muestra la Fig. 5.

La primera vez que se ejecuta el proceso, el estado del modelo se fija a OFF. Si el codec no soporta supresión de silencios se planificará una nueva interrupción un tiempo de trama T más adelante y la fuente quedará en estado ON durante el resto de la simulación. Cuando la supresión de silencios es soportada, si se produce un cambio de estado (p.ej. ON→OFF), las variables *Calcula_on* y *Calcula_off* se activan para calcular la duración del periodo ON u OFF respectivamente. Si el codec además incluye generación de tramas SID, necesitamos distinguir el caso AMR (tiempo entre tramas SID determinista) del caso G.729AB (tiempo entre tramas SID aleatorio). Este modelo también incluye algunas

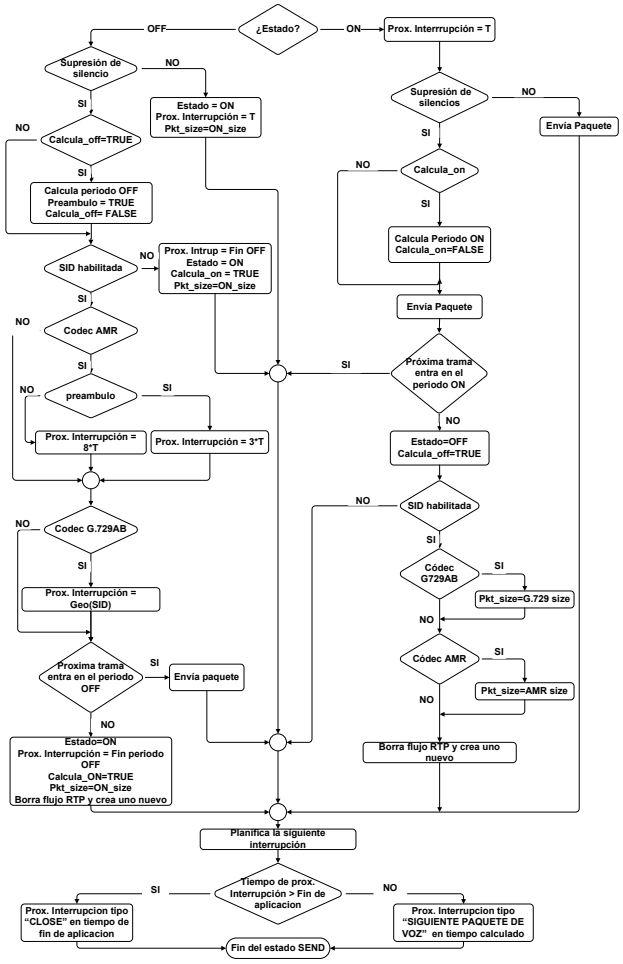


Fig. 5. Diagrama de flujo de nuevo estado *send*

mejoras que aplican tanto al modelo ON-SID creado como al modelo ON-OFF.

B. Mejora del modelo ON-OFF

El diagrama de flujo presentado en la Fig. 5 describe la generación de tráfico de una fuente de voz generalizada, por lo que debe ser válida para todo tipo de fuentes (CBR, VBR o VBR+CNG). Cuando la fuente es VBR (independientemente de que soporte CNG) el cálculo de la duración de los periodos ON y OFF se hace tradicionalmente utilizando una función de distribución exponencial [4]. El uso de una distribución continua en el tiempo aplicada a un proceso como el envío de tramas de voz, que es claramente discreto (se envían cada tiempo de trama T), provoca errores de modelado que puede subsanarse usando su equivalente discreta: la distribución geométrica. Además, si usamos una distribución continua una trama de voz puede ser enviada después de que finalice su periodo ON, lo que aumenta la duración real del periodo y provoca un pequeño incremento del régimen binario medio con respecto al teórico (ver Fig. 6). Teniendo en cuenta que las tramas deben enviarse en tiempos discretos (cada tiempo de trama T) tiene sentido forzar a los periodos ON y OFF a tener una duración que sea múltiplo de T. Este número entero es la salida de la distribución geométrica que coincide con la media de la distribución exponencial.

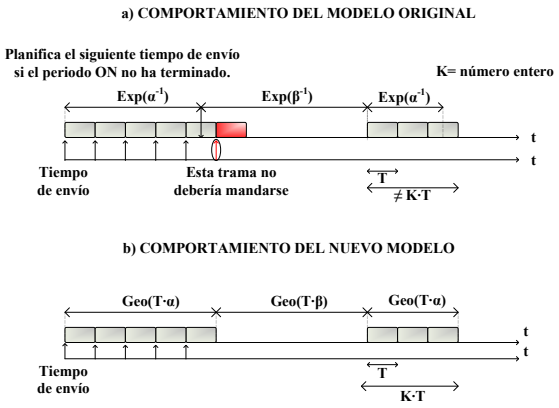


Fig. 6. Modificación de la distribución de los periodos ON y OFF

V. VALIDACIÓN

Para validar el comportamiento de nuestro nuevo modelo realizamos una serie de simulaciones que contrastamos con trazas de VoIP grabadas de una línea RDSI. Aunque es bien conocido que las trazas de voz no se ajustan totalmente con una distribución exponencial, su uso nos permitirá tener una idea de la precisión del modelo implementado y de la viabilidad de usar el simulador OPNET con codecs que incluyan CNG. Para obtener las trazas de voz hemos grabado un conjunto de conversaciones (un total de 500 minutos) desde una línea RDSI. Este material (conversaciones en formato *raw*) ha sido codificado usando los codecs G.729AB y AMR que poseen capacidad CNG, obteniendo un conjunto de ficheros de texto (ficheros *ftype*) que contienen una secuencia de identificadores $\{0,1,2\}$ que indican el tipo de cada trama codificada para cada instante de la conversación (voz, SID o nada respectivamente). Usamos esos ficheros *ftype* para dos propósitos: obtener valores realistas para los parámetros que usaremos en las simulaciones, y medir el régimen binario medio de cada conversación (referido como *trace* en nuestros resultados).

Para validar la agregación de fuentes de voz emulamos el proceso de multiplexión de N fuentes (usamos ficheros *ftype* de 3 minutos elegidos aleatoriamente) mediante un sistema de *token-bucket* con parámetros de tasa de *tokens* $C = \alpha\ N \cdot R$ y un tamaño de buffer de m. La formación de paquetes RTP/UDP/IP es anterior al proceso de multiplexión. La salida de nuestro sistema *token-bucket* es el porcentaje medio de paquetes perdidos (distinguiendo el porcentaje debido a paquetes SID y a paquetes de voz). Estos valores se identifican como *trace* en los resultados de agregación de tráfico.

En cuanto a las simulaciones de OPNET, hemos creado un escenario para probar nuestro modelo de una fuente. Hemos definido varias aplicaciones de voz que usan codecs diferentes y analizado las estadísticas de régimen binario medio (a nivel de aplicación) obtenidas de las simulaciones. Para simular el tráfico agregado hemos creado dos escenarios con 20 y 40 fuentes respectivamente. Usamos el protocolo *ppp* (point to point protocol) en la capa de enlace que permite una asignación libre de la capacidad del enlace de salida del nodo multiplexor (un *router* en este caso). La Fig. 7 muestra uno de los escenarios.

Para obtener resultados fiables realizamos una batería de

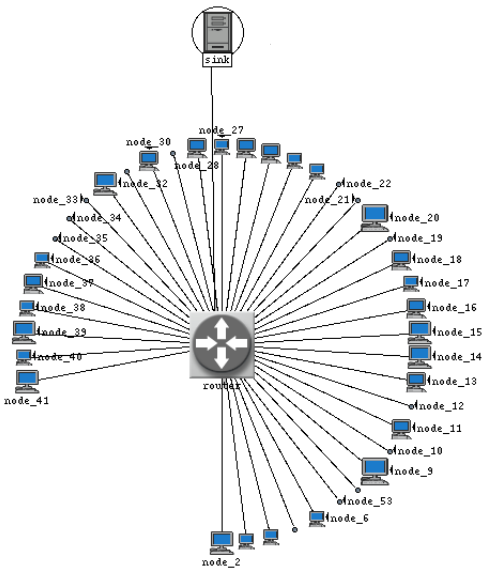


Fig. 7. Escenario de tráfico agregado de 40 flujos

20 simulaciones con diferentes semillas para cada escenario. En cada simulación la duración de la aplicación de VoIP es de 300 segundos.

A. *Resultados de las simulaciones para una fuente VoIP*

La Fig. 8 muestra los resultados obtenidos para el régimen binario medio de una fuente de voz que usa codecs con capacidad CNG (p.ej. AMR y G.729AB) y sin capacidad SID (G.729A soporta supresión de silencios pero no CNG). Como puede observarse, los resultados son distintos para cada simulación debido a la aleatoriedad de la duración de los periodos ON y OFF. Esto explica los grandes intervalos de confianza que se obtienen (ver Tabla I). Para el caso del codec G.729AB hemos usado una distribución exponencial para el tiempo entre tramas SID obtenida de las medidas experimentales.

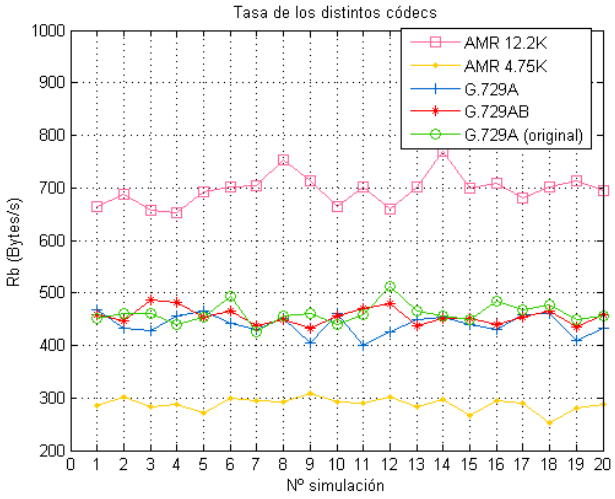


Fig. 8. Tasa media para distintos codecs y simulaciones

La Tabla I muestra los valores promedio de las 20 simulaciones así como las medias teóricas calculadas usando los valores de parámetros obtenidos de las trazas.

Tabla I
TASAS TEÓRICAS, SIMULADAS Y EXPERIMENTALES DE LOS CODECS EN BYTES/S

Codec	$\overline{R}$ teórico	$\overline{R}$ OPNET	I. +95%	I. -95%
AMR 4.75K	263	281	226.4	349.8
AMR 12.2K	675.5	696.3	529.9	862.7
G.729A (mejorado)	443	440.1	359.3	520.9
G.729A (original)	443	460.8	381.7	539.9
G.729AB	457.6	455.7	376.4	535
G.729AB (trazas)	456.5	—	—	—

Puede verse como el modelo ON-SID propuesto ofrece resultados más aproximados respecto a las tasas medias teóricas y a las trazas. Normalmente un codec G.729A genera menos paquetes que un G.729AB debido a la ausencia de SID del primero. Sin embargo, en los resultados de las simulaciones en OPNET, el modelo original G.729A genera una tasa media $\overline{R} = 460.8$ B/s., que es mayor que la de trazas, $\overline{R} = 456.5$ B/s. Como consecuencia, el modelo original reflejará un porcentaje de pérdidas muy superior al real. Esta inconcluencia se debe a la imprecisión del modelo original apuntado en la Fig. 6 y desaparece al hacer uso del modelo mejorado. Al usar este modelo para el G.729A, se observa que la tasa media obtenida, $\overline{R} = 440.1$ B/s., está mucho más cerca de la tasa teórica, $\overline{R} = 443$ B/s. que la ofrecida por modelo original, $\overline{R} = 460.8$ B/s. Por otra parte, la tasa media obtenida en las simulaciones del G.729AB, $\overline{R} = 455.7$ B/s., se encuentra muy cerca del valor teórico, $\overline{R} = 457.6$ B/s. y del experimental, $\overline{R} = 456.5$ B/s. Esto demuestra que se han realizado dos mejoras independientes en el modelo:

1) Ajuste del modelo de generación de tráfico ON-OFF existente.
2) Inclusión de la característica CNG.

B. *Resultado de simulación relativos al dimensionamiento de tráfico VoIP agregado*

En los escenarios de multiplexión descritos anteriormente se han realizado baterías de 20 simulaciones en las que se han utilizado valores distintos para el tamaño de buffer del *router* (m) y en la capacidad del enlace de salida (C). Como puede observarse en la Fig. 7, la topología de red es una estrella compuesta por estaciones de trabajo y un servidor. El nodo central es un *router* que interconecta las estaciones con el servidor. El *router* tiene una cola finita de tamaño m. Las aplicaciones usadas por las estaciones son las mismas que en la sección anterior, esto es: G.729A, G.729AB, AMR 4.75K y AMR 12.2K. La duración de las conversaciones es de nuevo 300 segundos. Para calcular la probabilidad de pérdidas usamos la relación entre el número total de paquetes de aplicación recibidos por el destino (todos las estaciones tiene el mismo) y el total de paquetes enviados al destino. Puesto que todos los paquetes tienen que atravesar el *router*, las pérdidas sólo se pueden producir en su cola de salida. OPNET ofrece el estadístico *overflow* que indica cada instante en el que sucede un desbordamiento en la cola. De esta estadística obtenemos el número de paquetes perdidos. Para averiguar el total de paquetes generados hacemos uso de la estadística

busy que muestra los instantes de recepción de paquetes en una cola.

Las Fig. 9 y 10 muestran la variación de la probabilidad de pérdidas con el tamaño de buffer para los agregados de 20 y 40 flujos. Se observa como disminuye la probabilidad de pérdidas a medida que aumenta el buffer del multiplexor. Los resultados muestran la precisión del modelo propuesto para el G.729AB con respecto a las trazas. Como puede verse, el modelo original ON-OFF sobreestima la probabilidad real de pérdidas. Las imprecisiones del modelo de generación de tráfico original de OPNET provocan que el codec G.729A, sin tramas SID, genere más tráfico que el G.729AB. Con un modelo más correcto, la tasa del codec G.729 sería menor a la del G.729AB y, por lo tanto, daría lugar a menos pérdidas. También hay que notar como disminuye la probabilidad de pérdidas conforme aumenta el número de fuentes. Las pérdidas se pueden producir cuando hay demasiadas fuentes en estado activo a la vez. En [13] se muestra cómo estas pérdidas debidas a la generación de tráfico a ráfagas se reducen a medida que N crece, explicando porqué las pérdidas son menores para el escenario con $N = 40$.

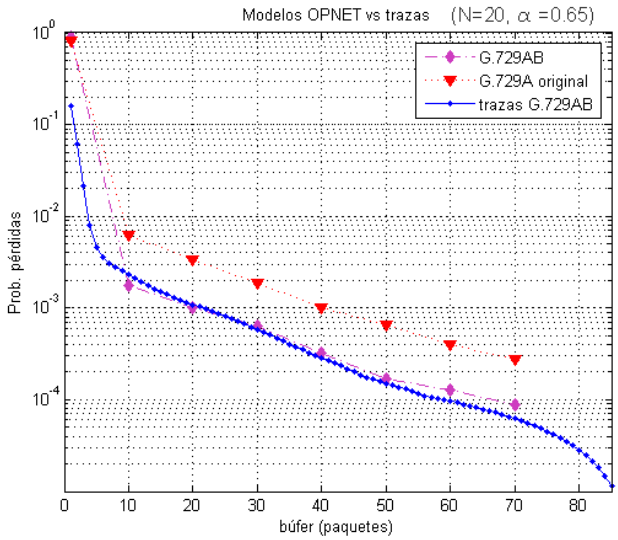


Fig. 9. Proceso de multiplexión de las trazas (G.729AB) frente a las simulaciones con OPNET para 20 fuentes.

Finalmente, la Fig. 11 muestra una aplicación del dimensionamiento con la que podemos conocer la probabilidad de pérdidas asociada al uso de distintas capacidades en el enlace de salida (expresado en términos $\alpha = C/(N \cdot R_{on})$). Para realizar este experimento, se ha fijado un valor de buffer tal que el retardo máximo en él sea de 50 ms. Ese valor se puede obtener de la siguiente expresión:

$$m = D \cdot \frac{C}{L} \tag{5}$$

donde m es el tamaño de buffer, D el retardo máximo en el buffer, C la capacidad de salida y L el tamaño del paquete. Por lo tanto, para cada α tendremos un valor distinto de m que fija $D = 50$ ms. De nuevo nuestro modelo ON-SID está más cerca de los resultados experimentales que el modelo original (probado para el codec G.729A).

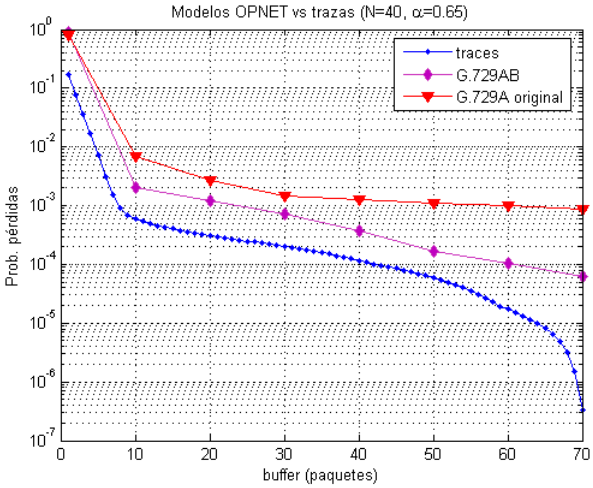


Fig. 10. Proceso de multiplexión de las trazas (G.729AB) frente a las simulaciones con OPNET para 40 fuentes

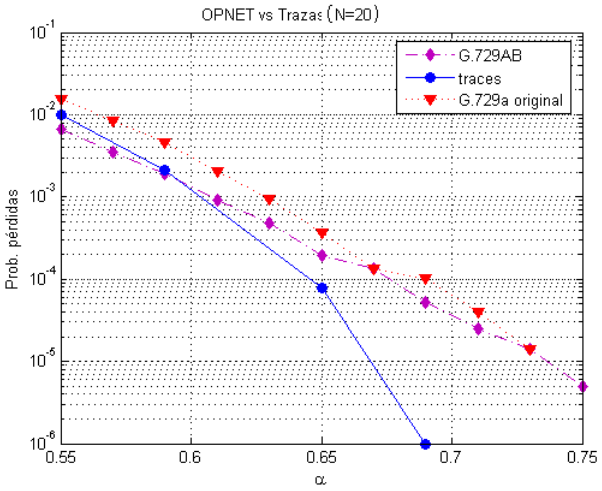


Fig. 11. Ejemplo de dimensionamiento expresado en α, para 20 fuentes

VI. Conclusiones

En este artículo hemos presentado un modelo que captura el patrón de tráfico de aquellos codecs que soportan CNG. Hemos implementado nuestro modelo en OPNET y validado los resultados con trazas experimentales. El modelo implementado mejora el modelo ON-OFF y permite obtener mejores resultados en el problema del dimensionamiento. Por tanto, nuestro modelo ON-SID nos permite tener una mejor caracterización del tráfico VoIP y hacer uso del simulador OPNET para validar futuras propuestas. En la actualidad se está trabajando para poner este nuevo modelo de VoIP a disposición de todos los usuarios de OPNET.

Agradecimientos

Parte de este trabajo ha sido desarrollado gracias a subvenciones otorgadas por el proyecto Minerva (1c-006) y por la Corporación Tecnológica de Andalucía (CTA).

Referencias

[1] Estepa A., Estepa R. and Vozmediano J.: 'A new approach for VoIP traffic characterization', in IEEE Communication Letters. Vol. 8, Issue 10. (2004) pp. 644-647.

[2] Estepa A., Estepa R. and Vozmediano J.: 'Paquetization and silence influence on VoIP traffic profil'. Lecture Notes on Computer Science 2899-Multimedia Interactive Protocols and Systems MIPS 2003.
[3] H.P. Sze, S.C. Liew, J.Y.B. Lee, and D.C.S. Yip.: 'A multiplexing scheme for H.323 voice-over-IP applications,'in IEEE J. Selcet. Areas Commun., vol. 20, pp. 1360-1368, Sept. 2002.
[4] Anick D., Mitra D., and Shondi M., 'Stochastic theory of a data-handling system with multiple sources', in Bell System Technical Journal. Vol. 61. (1982) pp. 1871-1894.
[5] Baiocchi A., Melazzi N., and Roveri A.:'Queueing performance and control in ATM', in Proceedings of the 13-th International Teletraffic Congress, Copenhagen (1991) pp. 13-18
[6] Salah K.: 'On the deployment of VoIP in Ethernet networks: methodology and case study', in Computer Communications. Vol. 29, Issue 8. (2006) pp. 1039-1054
[7] Baiocchi A. and Melazzi N.: 'Steady-State Analysis of the MMPP/G/1/K Queue, in IEEE Transactions on Communications'. Vol. 41, Issue 4. (1993) pp. 531-534
[8] Heffes H. and Lucantoni D.: 'A markov Modulated characterization of packetized voice and data traffic and related statistical multiplexer performance', in IEEE Journal of Selected Areas in telecommunication Communications. Vol. 4, Issue 6.(1986) pp. 856-868.
[9] Stewart R. et al.: 'Analysis of local internet enterprise networking multiplexer for ON-OFF VoIP sources', in Electronic Letters. Vol. 36, Issue 21. (2000) pp. 1825-1826
[10] Schormans J. et al.: 'Buffer overflow probability for Multiplexed on-off VoIP sources', in Electronic Letters. Vol. 36, Issue 6. (2000) pp. 523-524
[11] Bruno R., Garroppo R. and Giordano S.: 'Token bucket dimensioning for aggregate VoIP sources', in Proceeding of the IEEE ATM Workshop, 2000.
[12] Estepa A., Estepa R., Pacheco A.: 'Accurate Resource Estimation for Generalized Voip Sources'. Telecommunication Systems. Vol. 37. 2008. Pag. 1-14
[13] Jiang, W. and Schulzrinne, H.:,'Analysis of on-off patterns in VoIP and their effect on voice traffic aggregation', in Proc. ICCN 2000.

Propuesta de un protocolo de autenticación mejorado para IMS y estudio de viabilidad

Daniel Díaz-Sánchez, Davide Proserpio, Andrés Marín-López, Florina Almenárez-Mendoza, Alberto Cortés-Martín
Departamento de ingeniería telemática,
Universidad Carlos III de Madrid
Avda de la Universidad 30, E-28911, Leganés (Spain)
{dds,dproserp,amarin,florina,alcortes}@it.uc3m.es.

***Resumen*—El proceso estándar de autenticación de Internet Multimedia Subsystem (IMS) dispara dos procesos de autenticación, uno con la red de acceso y otro con la red Internet Multimedia Subsystem (IMS), o core del sistema pese a que en la mayoría de las ocasiones, las credenciales las verifica la misma entidad. Esto implica la introducción de un gran retardo en la fase de registro. Algunos trabajos aceleran el registro mediante túneles tipo TLS pudiendo sufrir un ataque de man-in-the-middle (MITM) si el túnel no tiene ningún tipo de relación criptográfica con el mecanismo de autenticación usado en el interior. En este artículo presentamos un protocolo de autenticación para IMS que permite ahorrar hasta 3 Round Trip Times. Tal protocolo tiene como objetivo mantener una seguridad equivalente al original permitiendo acelerar la autenticación para cualquier tipo de red de acceso y minimizar el impacto de su implantación. Por otro lado, permite reducir el proceso entero de autenticación a un solo round trip time en algunos escenarios. En el artículo se analiza la autenticación estándar en IMS y se hace un estudio de seguridad de la misma. Se describe una propuesta del protocolo de autenticación y se demuestra que mantiene el nivel de seguridad del proceso estándar, se analizan otras propuestas de autenticación en IMS y se lleva a cabo un estudio de viabilidad del protocolo propuesto para su implantación en sistemas comerciales como los basados en Symbian y J2ME**

***Palabras Clave*—IMS, Authentication, Security, Symbian, J2ME**

I. Introducción

Internet Multimedia Subsystem (IMS) es un conjunto de esfuerzos para definir el cambio desde las redes a conmutación de circuito a redes a conmutación de paquetes que tiene como aspecto fundamental la convergencia. Los servicios en IMS se proporcionan a los usuarios a través de una infraestructura que reutiliza parte de la estructura proporcionada por los servicios existentes, como por ejemplo subscripción o gestión de usuarios, facturación y QoS. Entre los procesos más críticos se ha identificado el de registro a la red IMS. Esta tarea se lleva a cabo cuando el ME (Mobile Equipement) quiere conectarse a la red IMS por medio de una red de acceso como puede ser WLAN, UMTS, LTE o WIMAX. Por eso es necesaria una doble autenticación: una con la red de acceso y una con la red IMS. Típicamente, debido a acuerdos o federaciones, las mismas credenciales son utilizadas en ambos procesos de autenticación causando un gran retardo. Algunas soluciones, como las analizadas en la sección V, permiten acelerar el proceso de registro para una red de acceso especifica combinando los dos procesos de autenticación en uno solo. Otras soluciones proponen utilizar un autenticación por medio de túneles genéricos pero son susceptibles de sufrir el ataque Man-in-the-middle (MITM) descrito en [1] y [2]. El algoritmo de registro de IMS, definido en [3], provee autenticación mutua y generación de clave para establecer un asociación segura entre la red IMS y el ME (con IPSEC). Este proceso requiere redirigir los mensajes de registro hacia la red local del ME cuando éste se encuentra en roaming de modo que se puedan comprobar las credenciales, introduciendo un retardo total de dos round trip time que, al añadirlo al tiempo empleado en la autenticación con la red de acceso (2 RTTs), da un resultado final de 4 RTTs. Para solucionar dicho problema, este artículo presenta un protocolo de autenticación que, en algunos escenarios, permite reducir el proceso a un solo RTT para el registro a la red IMS. Este protocolo permite realizar una autenticación completa (red IMS y red de acceso) relacionando las dos autenticaciones por medio de un token; también permite resumir un registro IMS previo utilizando material de clave anterior. De este modo, como se demostrará en la sección III-B, el protocolo de registro de IMS propuesto es resistente al mencionado ataque MITM pese al uso de túneles en algunos escenarios. En términos criptográficos este proceso vincula ambas autenticaciones (de red e IMS) y genera también material de clave que puede ser usado en un asociación segura (IPSEC) entre el ME y el P-CSCF. En algún escenario esta mejora de la fase de registro en IMS puede ser combinada con **tuneles EAP-TLS** y **TLS extractor** [4] permitiendo ahorrar hasta tres RTTs. Esta propuesta ha sido diseñada para mantener el registro seguro, minimizar el impacto en la fase de despliegue y evitar el ataque MITM. Es más, el análisis de seguridad mostrará que este protocolo provee el mismo nivel de protección que un autenticación estándar. Para finalizar, este artículo analizará algunas implementaciones comerciales de la pila de protocolo IMS para determinar las posibilidades de despliegue de la propuesta. Este artículo es organizando de la siguiente manera, II introduce el problema del mecanismo de la autenticación compuesta, la sección III trata de IMS, su registro estándar y un análisis de seguridad de la misma. La sección IV describe la nuestra propuesta para reducir el tiempo de registro por medio de una prueba criptográfica de autenticación que vincule ambos protocolos. Además, se analiza el protocolo de reanudación (o resumen), mostrando como éste provee el mismo nivel de seguridad que un registro estándar. La sección V analiza algunas propuestas que implementan una autenticación a la red IMS sin la utilización de pruebas criptográficas. En la sección VI se describie la experiencia adquirida al implementar el protocolo propuesto en soluciones

comerciales. La sección VII resume el trabajo realizado.

II. El problema de la autenticación compuesta

Los protocolos de autenticación de UMTS e IMS son independientes uno del otro y sus modulos de subscripción, USIM y ISIM respectivamente, gestionan material de clave diferente. Dicho material de clave puede ser reutilizado por la red de acceso en fase de autenticación, evitando así que el UICC implemente un módulo de subscripción por cada nueva tecnología. Esta simplificación hace que sea posible un ataque MITM ya que el algoritmo utilizado no tiene forma de saber el propósito de la autenticación. Por ejemplo el MITM puede utilizar los mensajes de registro a una red de acceso para suplantar un usuario y de este modo obtener acceso a la misma u otra red que espera las credenciales conseguidas. La única forma de evitar este tipo de ataque consiste en que se obtenga una clave (o unas claves) desde el mecanismo de autenticación y que sea utilizada para proteger el canal entre el ME y el Network Autentication Server (NAS).
Hoy en día se pueden encontrar varios trabajos que proponen como solución la autenticación a través de un túnel que permite el transporte de mensajes EAP sobre otros protocolos. La idea es reutilizar protocolos de autenticación existentes para crear un túnel con la red de acceso y así autenticar el NAS, para luego empezar el registro con el protocolo interno. De este modo, cuando el NAS es autenticado, envía los mensajes de autenticación del protocolo interno hacia el modulo de autenticación del servicio (por ejemplo el HSS en IMS). Esto permite que los mensajes de registro al servicio sean seguros porqué se envian sobre el túnel creado entre ME y NAS. Además utilizando un túnel con la red de acceso es posible mover la fase de autenticacíon a niveles superiores al nivel 2 por ejemplo utilizando Protocol for carrying Authentication for Network Access (PANA). Aún así es posible un ataque MITM dado que estos tipos de túneles requieren una distribución de credenciales a cada NAS que se utilice. El problema explicado en [1] y [2] aparece cuando un protocolo de autenticación esta diseñado como combinación de dos protocolos: un protocolo interno y uno externo. El protocolo externo, como puede ser TLS [5], protege el intercambio de mensajes del protocolo interior que puede llegar a enviar las credenciales en claro. Este último es utilizado para autenticar el usuario con la red mientras que el protocolo externo es utilizado para autenticar la red con el usuario y proteger el intercambio de datos. Entre los protocolos afectados por este ataque encontramos PEAP, EAP-TLS, PIC y PANA sobre TLS [6]. El problema aparece bajo alguna de las siguientes condiciones.

1. El protocolo interno puede ser utilizado en otros entornos, como HTTP Basic Auth. Esto pasa cuando el protocolo interno no tiene modo de saber si esta siendo utilizado dentro de un túnel o no y por tanto un atacante puede hacer un reply sin que el protocolo interior sea consciente.
2. El cliente falla al comprobar el certificado del servidor en el protocolo exterior. Esto puede ser frecuente dado que durante un proceso de conexión a la red el ME no tiene conexión a internet para descargar Certificate Revocation Lists (CRLs) o cualquier otra información necesaria para comprobar los certificados. Además, a pesar de que eso es un error inaceptable por parte del cliente, la red debe de proporcionar mecanismos para evitar que un único error del cliente pueda comprometer la seguridad (especialmente cuando usuarios no profesionales están involucrados).

El ataque funciona de la siguientes manera: MITM espera hasta que un dispositivo legitimo (ME) empiece el proceso de autenticación sin el utilización de un túnel. Entonces, el MITM crea un túnel con la red de acceso y sobre este empieza a enviar mensajes de autenticación hasta que el cliente legítimo sea autenticado. En ese momento, MITM obtiene las claves para proteger el canal utilizando las claves del túnel externo, robando así el servicio al cliente legítimo. Para evitar ese tipo de ataque existen dos aproximaciones. Una de ellas consiste en que el protocolo interior proporcione, además de autenticación, unas claves. Esas claves deberían de ser utilizadas para proteger el canal entre el cliente y el servidor. Por lo tanto, hay una autenticación implícita porque solo el cliente conoce estas claves. Esto se soluciona con la segunda aproximación, en la que las claves del túnel externo son obtenidas desde un secreto de larga duracción usado en el protocolo interno o si ambos protocolos están relacionados de alguna manera.

III. Autenticación en IMS

IP Multimedia subsystem (IMS) proporciona servicios utilizando servidores llamados Call Session Control function (CSCF) y el el protocolo SIP (Session Initiation Protocol) [7]. Los datos de subscripción son manejados por el Home Subscriber Server (HSS) y el Authentication Center (AuC). IMS define los siguientes tipos de CSCF: un proxy-CSCF (P-CSCF), normalmente localizado en la red visitada que se encarga de redireccionar los mensajes SIP desde el ME hacia la red local dado que solo el HSS puede autenticar el usuario. Además, el P-CSCF establece un asociación segura con los MEs, utilizando IPSec, permitiendo así que se pueda controlar la integridad y autenticidad de los mensajes. Las claves necesarias para establecer tal asociación, C_K and I_K, se obtienen del resultado de la autenticación realizada con el HSS.
I-CSCF o interrogating CSCF, localizado en la red local que tiene como objetivo localizar el servidor encargado de gestionar los mensajes de autenticación enviados por los MEs y S-CSCF (serving-CSCF) autentica los usuarios pidiendo al HSS las informaciones de subscripción y los vectores de autenticación.

III-A. El protocolo de registro de IMS

El mecanismo de autenticación utilizado en IMS está basado en la autenticación HTTP Digest [8] y en el algoritmo AKAV1-MD5 [3], que requiere un intercambio de 4 mensajes (2RTT) entre el usuario, en la red visitada, y su red de acceso. La autenticación IMS está basada en el UICC (Universal Integrated Circuit), una tarjeta localizada en el ME que comparte un secreto de larga duracción con el HSS. La fase de registro en IMS funciona de ese modo (se vea también Fig. 1):

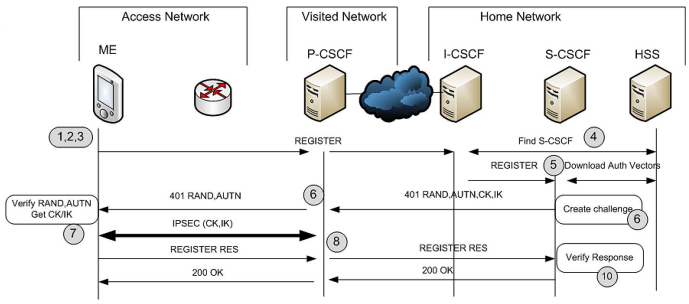


Figura 1. Intercambio de mensajes para un registro con la red IMS

1. El ME se registra con la red de acceso de algún manera de modo que puede obtener conexión IP y así descubrir el P-CSCF.
2. El ME utiliza los datos de la UICC para componer un mensaje SIP REGISTER que incluye el URI de registro (para localizar la red local), la identidad publica y privada y la dirección del contacto. Además, incluye la cabecera Security Client para indicar al P-CSCF el algoritmo suportado para la creación de la asociación IPSEC.
3. El ME envía el mensaje REGISTER al P-CSCF. El P-CSCF obtiene la dirección del I-CSCF en la red local, añade la cabecera P-Visited-Network al mensaje y lo redirige al I-CSCF.
4. El I-CSCF envía un mensaje diameter User-Authentication Request (UAR)al HSS que contiene las identidades publica y privada del usuario para descubrir el S-CSCF que se encargará del mensaje de REGISTER.
5. El S-CSCF contacta con el HSS para que le envie los vectores de autenticación. Estos vectores contienen un reto aleatorio (RAND), un token de autenticación (AUTN), la respuesta esperada (XRES), una clave de integridad (I_K) y una clave de confianza (C_K). AUTN es creada por el HSS utilizando un secreto de larga duracción que comparte con el ME y un numero de secuencia (SQN). Luego, el S-CSCF compone un mensaje 401 Unauthorized que contiene la cabecera WWW-Authenticate con la representación en base 64 de AUTN y RAND (parametro nonce). Además incluye en el mensaje I_K y C_K para el P-CSCF. Una vez compuesto, el S-CSCF envía el mensaje al ME, a través del I-CSCF y del P-CSCF.
6. El P-CSCF extrae del mensaje 401 Anouthorized C_K y I_K y añade la cabecera Security-Server con uno de los algoritmos propuestos por el cliente para crear la asociación IPSEC.
7. El ME extrae, con la ayuda de la UICC, AUTN, RAND, I_K y C_K de la cabecera WWW-Authenticate. Calcula la respuesta al nonce recibido (RES) y establece un asociación segura con el P-CSCF utilizando las claves I_K/C_K. Entonces, el ME compone un nuevo mensaje REGISTER que contiene la respuesta calculada y la cabecera Security-Verify creada copiando la cabecera Security-Server enviada por el P-CSCF, y envía el mensaje al P-CSCF sobre la nueva asociación segura creada.
8. El P-CSCF una vez recibido el mensaje puede implícitamente autenticar el ME (porqué el mensaje ha sido recibido bajo un asociación segura con I_K y C_K). Controla la cabecera Security-Verify y se asegura que nadie ha cambiado su contenido. Hecho esto, redirige el mensaje al I-CSCF en la home network.
9. El I-CSCF envía el mensaje al S-CSCF previamente asignado.
10. EL S-CSCF recibe el mensaje y controla el valor RES. Si este valor es igual al XRES, puede estar seguro de que el usuario es legitimo. Así que, crea un nuevo mensaje SIP, 200 OK, y lo envía al ME, a través del I-CSCF y del P-CSCF. Aquí concluye el proceso de registro.

III-B. Análisis de seguridad del registro en IMS

El protocolo propuesto tiene que proporcionar el mismo nivel de seguridad que el de un registro IMS estándar. Por esta razón se realizará un análisis de seguridad del protocolo estándar. Las condiciones iniciales consisten en que el ME y el HSS comparten un secreto de larga duracción K.

$$\text{ME} \Leftrightarrow_k \text{HSS}$$

Paso 3 : El P-CSCF incluye la cabecera P-Visited-Network en el REGISTER proporcionando su identidad a la red local. El mensaje REGISTER es transmitido sobre una interfaz segura (Za).

$$\text{S-CSCF confia (P-CSCF dice REGISTER)}$$

Paso 6 : El P-CSCF recibe I_K y C_K desde el S-CSCF bajo una red segura entre proveedores.

$$\text{P-CSCF confia (S-CSCF dice } I_K, C_K)$$

Pas 7 (A) : EL ME extrae I_K, C_K, AUTN y RAND desde la cabecera WWW-Authenticate, así que puede **autenticar la red local**.

$$\text{Desde que ME} \Leftrightarrow_k \text{HSS y}$$
$$\text{ME ve } \{I_K, C_K, AUTN, RAND\} \text{ desde } \{I_K, C_K, AUTN, RAND\}_K$$
$$\textbf{entonces ME confia (S-CSCF dice } \{I_K, C_K, AUTN, RAND\}_K).$$

Paso 7 (B) : EL ME confia en que el P-CSCF es seguro por el hecho de que su red lo acepta como valido. Paso 7 (C) : El ME crea un asociación segura con el punto X utilizando I_K y C_K. Dado que C_K y I_K vienen proporcionados por la red del ME hacia un P-CSCF autorizado, el ME esta seguro de que el punto X es el P-CSCF.

$$\text{Desde que ME confia (ME} \Leftrightarrow_{C_K, I_K} \text{P-CSCF) y}$$
$$\text{ME ve } \{\text{IPSEC-Payload}\}_{C_K, I_K}$$
$$\textbf{entonces ME confia (P-CSCF dice IPSEC-Payload).}$$

Paso 8: El P-CSCF recibe el mensaje REGISTER con la respuesta al reto sobre un asociación segura así que el ME es implícitamente autenticado por el P-CSCF.

$$\text{Desde que ME confia (ME} \Leftrightarrow_{C_K, I_K} \text{P-CSCF) y P-CSCF ve}$$
$$\{REGISTER\}_{C_K, I_K}$$
$$\textbf{entonces P-CSCF confia (ME dice REGISTER).}$$

Paso 10: El S-CSCF controla el mensaje con la respuesta del ME comparando RES con XRES. Si ambos son iguales, el S-CSCF **autentica el ME**. El análisis de seguridad es la misma que por el punto 7 (A).

El protocolo de registro de IMS es resistente al ataque MITM descrito en la sección II por el hecho de que las claves

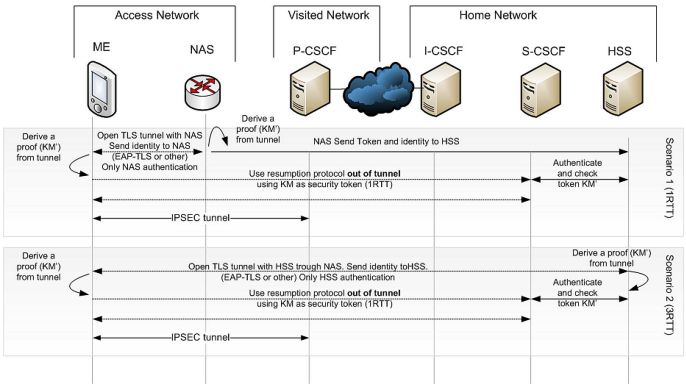


Figura 2. Protocolo de reanudación con prueba criptográfica

obtenidas durante la autenticación en el protocolo interno son utilizadas como material de claves en la asociación segura entre ME y P-CSCF.

IV. El protocolo de registro combinado

Los objetivos del protocolo de registro combinado son: reducir el tiempo de registro sin comprometer el grado de seguridad, hacer su utilización posible por cualquier tipo de tecnología existente o nueva y ma ntener compatibilidad con la actual fase de autenticación.
Por poder satisfacer tales objetivos, el protocolo diseñado tiene que cumplir algunos requisitos: 1)La autenticaión con la red de acceso tiene que ser posible a cualquier nivel, por eso el registro tiene que utilizar EAP que pude ser usado a nivel 2 o cualquier nivel superior (utilizando PANA para transportar EAP). 2) El ME tiene que ser capaz de autenticarse con el NAS y poder extraer material criptográfico para el registro con IMS. Además, el nivel 2 tiene que estar protegido. Esto supone que a través del método EAP se deriven claves (EAP-TLS) y material criptográfico (TLS EXPORTER [4]). 3) El registro con IMS tiene que estar relacionado criptográficamente con la autenticación con la red de accesso. 4) La asociación segura entre ME y P-CSCF tiene que ser protegida por medio de una clave de largo término (K_I) y la "prueba" extraida de la autenticación con la red de acceso.
Los escenarios donde se puede utilizar nuestra propuesta se muestran en la Fig. 2.

En el primer escenario el ME abre un túnel, utilizando EAP-TLS sobre L2 o PANA, con el Network Authentication Serivce (NAS) y proporciona su identidad de modo que el NAS puede resolver la direccion de la red local. El NAS extrae el material de clave del túnel por medio de TLS extractor y de ahí obtiene P_N y P_I, aplicando una Pseudo Random Function (PRF) sobre la concatenación de la Master Key con dos cadenas de texto diferentes. El ME también extrae el material de clave y obtiene las pruebas.El ME empieza el proceso de autenticación,incluye P_N, como token, en el mensaje REGISTER protegido por una firma (que puede ser comprobada por el HSS). En el mismo momento el NAS envía P_N y P_I al HSS utilizando el protocolo DIAMETER. De este modo, la firma puede utilizarse para autenticar el ME y P_N para relacionar la red de acceso con la red IMS. Como último paso, el HSS proporciona las claves ($C'_K = PRF(C_K|P_I)$ y $I'_K = PRF(I_K|P_I)$) para la asociacion segura IPSec. Tales claves, permiten relacionar ambos procesos de autenticación y así evitar un posible ataque MITM. De este modo, el HSS puede explicitamente autenticar el ME (con la firma), el ME autentica implicitamente el NAS, y este autentica implicitamente el ME. Este proceso ahorra 3 RTT sin comprometer la seguridad y evitando el ataque MITM. En el segundo escenario, el túnel EAP-TLS se abre directamente con el HSS, ahorrando un único RTT.
En el diseño del protocolo se cuenta con una clave asimetrica que tiene que ser registrada por el ME antes de poder ser utilizada. Además, esta clave tiene que permitir la generación de firmas. Los proveedores deben imponer una mínima longitud y un máximo tiempo de uso de esta por medio de una política. La clave, llamada R_k (clave de reanudación), debería de ser guardada en el HSS junto con el token proporcionado por la red de acceso.

IV-A. Definición del protocolo combinado y análisis de seguridad

En esta sección se describe el protocolo de registro detalladamente. La descripción siguiente asume que el usuario ha generado previamente las claves pública y privada y ha registrado la pública (R_k) bajo su perfil en el HSS.
El ME abre un túnel EAP-TLS con el NAS (1^{st} escenario) o con el HSS (2^{st} escenario) a través de PANA o el protocolo L2. Una vez que HSS y ME hayan obtenido las pruebas criptográficas P_N y P_I, el ME ejecuta el procese de registro como explicado a continuación (se vea también Fig.3):

1. El ME compone un mensaje REGISTER, como se describió en la sección III-A incluyendo la cabecera Authorization y la cabecera Security-Client.
2. El ME incluye el campo **nonce** en la cabecera Authorization y también P_N (obtenido del túnel con el NAS) que será utilizado para relacionar las dos autenticaciones. Añade también un *auth-param* con el texto *"resume@idx"* que indica el índice de la clave R_K que tiene que utilizarse. Hecho esto, el ME genera un mensaje S/MIME incluyendo el en el cuerpo información sobre las cabeceras como explicado en Authenticated Identity Body Format [9], excepto por la inclusión, dentro del cuerpo y del cuerpo formado, de las cabeceras Authorization y Security-Client.
3. El ME envía el mensaje REGISTER fuera del túnel al P-CSCF. EL P-CSCF inserta el identificador P-Visited-Network y envía el mensaje al I-CSCF (que se encuentra en la red local del ME).
4. El I-CSCF descubre el S-CSCF y le envía el mensaje.
5. El S-CSCF extrae el **índice de clave** y el **nonce** desde las cabeceras. Contacta con el HSS para descargar los tokens (P_N y P_I), la clave de reanudación correspondiente al índice de clave y el nuevo vector de autenticación. EL S-CSCF compone un mensaje 401 Unauthorized que contiene la cabecera WWW-Authenticate con el nonce y incluyendo los valores en base64 de AUTN y RAND obtenidos del **nuevo vector de autenticación**. Entonces, el S-CSCF averigua la firma y si el nonce contiene el mismo P_N que ha recibido del NAS. Si ambos parámtros son validos, el S-CSCF autentica el usuario. Para informar al ME de que la autenticación ha tenido éxito, el S-CSCF

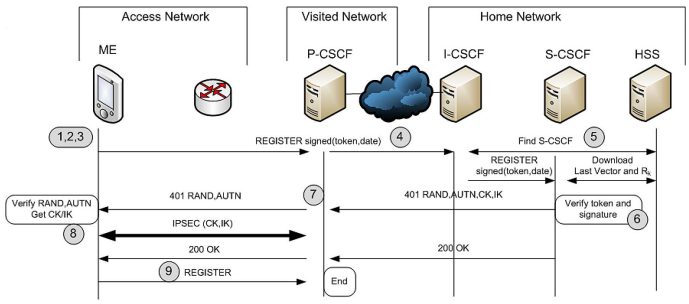


Figura 3. Intercambio de mensajes para un registro con combinado

Autenticación	Registro estándar	Registro con re-anudación
La red local autentica la red visitada (federación)	paso 3	paso 4
La red visitada autentica la red local (federación)	paso 6	paso 7
La red local autentica el ME (explicito)	paso 10	paso 6
El ME autentica la red local (explicito)	paso 7A	paso 8A
El ME autentica la red visitada (implicito)	paso 7b-7C	paso 8B-8C
La red visitada autentica el ME (implicito)	paso 8	paso 9

Cuadro I
AUTENTICACIÓN ENTRE LAS ENTIDADES INVOLUCRADAS DURANTE EL PROCESO DE AUTENTICACIÓN PARA AMBOS PROTOCOLOS

incluye en la cabecera WWW-Authenticate el parámetro *auth-param* con el valor *"resume@idx"*. Además el S-CSCF obtiene las dos claves para la asociación IPSec, $I'_K = PRF(I_K|P_I)$ y $C'_K = PRF(C_K|P_I)$ y las incluye en el mensaje. El S-CSCF envía el mensaje al ME, a través del I-CSCF y del P-CSCF. Hecho esto, compone otro mensaje, el 200 OK que será enviado inmediatamente después del mensaje 401 Unauthorized. De otra forma, si la fase de control no finaliza con éxito el S-CSCF envía sólo un mensaje 401 Unauthorized sin alguna modificación respecto a un registro estándar.

6. El P-CSCF extrae I'_K y C'_K desde el mensaje 401 Unauthorized, añade la cabecera Security-Server eligiendo un algoritmo para crear una asociación segura con el ME.
7. El ME extrae AUTN,RAND, I_K y C_K desde al cabecera WWW-Authenticate **autenticando así la red local**. Si la cabecera WWW-Authenticate contiene el parámetro *auth-param*, el ME obtiene $I'_K = PRF(I_K|P_I)$ y $C'_K = PRF(C_K|P_I)$. Establece una asociación segura con el P-CSCF y espera hasta que reciba el mensaje 200 OK. En caso de que el proceso fallara por error en la cabecera, su comportamiento es el mismo que en un proceso de registro estándar, calcula la respuesta RES y la incluye en un nuevo mensaje REGISTER.
8. Si la reanudación a sido aceptada por el S-CSCF (el mensaje contiene el auth-param), el ME crea un mensaje REGISTER que contiene sólo la cabecera Security-Verify y la cabecera **TO** indicando el P-CSCF (este mensaje no tiene que llegar a la red local). El ME envía el REGISTER a la dirección del P-CSCF sobre la nueva asociación segura creada con el mismo.

IV-B. Consideraciones de seguridad

En esta parte se analiza el protocolo propuesto como hemos hecho para el protocolo de registro de IMS. Este análisis demostrará cómo nuestra propuesta es tan segura como el registro estándar. Las condiciones iniciales son: el ME y el HSS comparten un secreto de larga duracción llamado K_I, una clave de reanudación R_k y dos tokens de seguridad (P_I and P_N):

ME $\Leftrightarrow_{K_I,R_k,P_I,P_N}$ HSS
HSS confia (ME tiene jurisdicción sobre R_k)

Paso 4 : El P-CSCF incluye el campo P-Visited-Network en el REGISTER confirmando su identidad a la red local del usuario. El mensaje REGISTER viene transmitido sobre un interfaz segura entre proveedores (como en el registro estándar, paso-3). Paso 6: El S-CSCF recibe el mensaje REGISTER enviado por el ME, descarga las claves R_k,P_I,P_N y el nuevo vector de autenticación. El S-CSCF primero controla la fecha del body firmado y el **nonce** con respecto a la fecha en el servidor. Luego controla si el P_N es el mismo que el recibido. Si fecha y token son validos, controla la firma contra R_k. Si la firma es valida, y la fecha y P_N también lo son, el S-CSCF autentica el usuario.

HSS confia (ME tiene jurisdicción sobre R_k) and
S-CSCF confía (ME confía nonce[1])
entonces S-CSCF confia en el nonce.

S-CSCF confia (ME dice nonce) y
S-CSCF confia en que el nonce esta actualizado
entonces S-CSCF confia (ME confia en el nonce).

Paso 7 : El P-CSCF recibe I'_K y C'_K desde el S-CSCF sobre una red segura entre proveedores (como en el registro estándar, paso-6).
Paso 8 (A) : El ME es capaz de extraer I'_K, C'_K, AUTN y RAND desde la cabecera WWW-Authenticate, de modo que puede **autenticar la red local** (como en el registro estándar paso-7A). Paso 8 (B) : El ME confía en que el P-CSCF es valido por el hecho de que su red lo acepta como tal.
Paso 8 (C) : El ME crea una asociación segura con el punto X utilizando C'_K y I'_K. Por el hecho de que C'_K y I'_K han sido proporcionadas por la red de acceso del ME a un P-CSCF de confianza, el ME esta seguro de que el punto X es el P-CSCF (como en el registro estándar paso-7C). C'_K y I'_K dependen de P_I, que es obtenido del túnel con el NAS, por eso no puede ser un MITM.
Paso 9: El P-CSCF recibe el mensaje REGISTER con la cabecera Security-Verify por medio de la asociación segura creada con el ME de modo que el ME es implícitamente autenticado por el P-CSCF.
La tabla 1 resume ambos procesos de los protocolos de autenticación indicando cuales entidades son explicitamente o implícitamente autenticadas.

[1]Este nonce es la representación en base 64 del token de seguridad.

V. Analisis de otras propuestas

[10] propone una solución para un autenticación segura, en un escenario donde el acceso es via wireless, basado en las credenciales almacenadas el la UICC. Esta solución requiere un P-CSCF en la red de acceso llamado WLAN P-CSCF. Este módulo se encarga de redirigir el mensaje REGISTER del ME hacia la red visitada y luego hacia la red local. El WLAN P-CSCF añade una cabecera indicando que tipo de autenticación esta pidiendo el ME: un única autenticación para la red de acceso y la red IMS o un acceso solo con la red wireless. Esta nueva cabecera no esta protegida y esto puede comprometer la seguridad. Además, esta solución propone que el WLAN P-CSCF sea utilizado como punto de conexión para la asociación segura con el ME utilizando las claves I_K y C_K en vez de utilizar el P-CSCF de la red visitada. Esto supone un aumento de los problemas de seguridad dado que la red local tiene que permitir que su material de claves sea utilizado en la red de acceso. Y hay más, esta solución requiere una fuerte relación de confianza entre los operadores y la red de acceso por el hecho de que el P-CSCF de la red visitada tiene que aceptar trafico proveniente del ME sin utilizar alguna asociación segura con él (en nombre del WLAN P-CSCF). En [11] los autores proponen una autenticación de un sólo paso para obtener acceso a la red IMS a través de un acceso a la red GPRS. Estos proponen un cambio en el SGSN de modo que se modifique cualquier mensaje de register incluyendo el IMSI asociado con la conexión, una vez que el ME ha activado un contexto PDP. El S-CSCF solicita el IMPI asociado con elIMSI, y si éste coincide con él de la petición considera el usuario autenticado. Este método utiliza una característica de una especifica red de acceso y por eso no puede ser considerado como una solución de carácter general. Además, la autenticación hacia la red IMS viene implementada sin criptografía, así que la fuerza de este algoritmo es proporcional a la seguridad de la red del operador. Cabe destacar que cada usuario puede impersonar otro sólo cambiando el IMSI. Otras soluciones [12] proponen mover la autenticación al nivel 2 de la pila OSI utilizando 802.11x en conjunto con EAP-AKA, quitando así la autenticación a nivel de servicio. Este tipo de solución no es independiente de la tecnología de acceso de modo que requiere definir un procedimiento específico por cada nueva tecnología que introduce modificaciones a nivel de servicio. Esto hace que el proceso de registro sea mas complicado y propenso a errores.

VI. Analisis de despliegue

Hemos analizados dos implementaciones comerciales de la pila de protocolo IMS para testar la compatibilidad con el popular IMS core y evaluar la viabilidad de implementación del protocolo propuesto. Hemos analizado las APIs y testado si soportaban IPSEC y S/MIME. Hemos utilizado OpenIMSCore (Instituto Fraunhofer) para hacer nuestros test dado que es un herramienta largamente utilizada y implementa correctamente las especificaciones IMS como demostrado en [13].

VI-A. Analisis de la implementación en Symbian

Hemos utilizado un móvil Nokia N95 que tiene instalado el sistema operativo Symbian 9.2 junto con el software de implementación S60 tercera edición (Feature Pack 1). Symbian proporciona una clase para gestionar los mensajes del registro en IMS llamada *CSIPHttpDigest*. Cuando un mensaje 401 Unauthorized es recibido, una función de callback, *ChallengeReceived*, viene llamada por la pila IMS. Para proporcionar una respuesta al challenge, el método *SetCredentialsL*, parte de la clase *CSIPHttpDigest*, tiene que ser llamado con las credenciales apropiadas. Por lo que concierne la señalización, este pila de protocolo IMS tiene un comportamiento diferente comparado con la mayoría de los clientes IMS ya que incluye en la cabecera Request-URI el numero de puerto del proxy. Aunque la RFC 3261 [7] dice que el URI puede opcionalmente incluir el puerto de destino del mensaje, es un operación poco frecuente, de modo que es necesario un arreglo para que OpenIMS pueda suportar el mensaje con la cabecera así creada. Además, el entero proceso de autenticación en esa implementación del protocolo IMS viene realizada por completo por el sistema operativo haciendo imposible la manipulación de la cabecera WWW-Authorization (no hay un nivel bajo para poder acceder a los mensajes SIP). Por lo que es la parte de seguridad, Symbian proporciona soporte para la creación de un cuerpo S/MIME por medio de OpenC que contiene una versión de libcrypto con funciones para una codificación simétrica y asimétrica y para hacer el hash de las contraseñas. OpenC provee también una versión de OpenSSL que soporta SSL,TLS y S/MIME. IPSEC puede ser configurado por medio de un API y además hay un cliente VPN.

VI-B. Analisis de la implementación en J2ME

Hemos implementado y testado un cliente IMS J2ME utilizando algunos emuladores cuales SDK S60 3rd FP1 MIDP y el *"Wireless Toolkit"*, además hemos utilizado dispositivos reales cuales Nokia E61 y Nokia N95. Por lo que concierne las APIs, J2ME utiliza la clase *SipConnectionNotifier* para enviar, recibir mensajes SIP y para analizar las cabeceras de los mensajes enviados por medio de una conexión que puede ser dedicada o compartida, creada con *SipClienConnections*.La conexión compartida comparte la identidad del usuario SIP y el puerto de escucha con las otras aplicaiones pero el entero proceso de registro viene llevado a cabo por el sistema. Mientras que una conexión dedicada no es compartida con las otras aplicaciones siendo la aplicación capaz de gestionar el proceso de registro. De toda forma, ninguno de los mencionados móviles nos ha permitido crear una conexión en modo dedicado. Hablando de señalización, J2ME, como Symbian, incluye el puerto destino en la cabecera Request-URI. Además de esto, otras diferencias en los mensajes de señalización han hecho necesarias algunas modificaciones para que todo funcionara: el emulador Wireless-Toolkit enviaba varios mensajes REGISTER desde diferentes puertos UDP en la fase de registro y esperaba una respuesta en el puerto enviado en el ultimo register; el SDK S60 3rd FP1 MIDP incluía un parametro *opaque* vacío en la cabecera WWW-Authenticate cunado componía la respuesta al challenge de modo que la cabecera venia considerada malformada. En lo referente al soporte de seguridad, J2ME proporciona algunos paquetes para este propósito e incluso las librerías IAIK pueden ser utilizadas para la creación del cuerpo S/MIME y de las claves.

Por ultimo, no hay algún tipo de soporte IPSEC excepto por [13].

VII. Conclusiones

Este artículo describe unas mejoras en el mecanismo de registro de IMS que permiten utilizar token de seguridad para poder relacionar la autenticación a la red de acceso y la autenticación a IMS o para reanudar una previa autenticación. El protocolo propuesto ha sido diseñado para poder ser utilizado en cualquier tipo de tecnología de acceso que existe o que se implemente ya que la autenticación a la red de acceso es posible a nivel 2 o a través de PANA. Además, permite evitar numerosos ataques porqué no utiliza las credenciales contenidas en la UICC directamente, sino que todo el proceso de registro depende de la correcta autenticación con la red IMS. Hemos analizado el nivel de seguridad de un registro estándar del protocolo IMS y de nuestra propuesta mostrando que ambos proporcionan el mismo nivel de seguridad y que autentican cada entidad involucrada en el proceso en el mismo modo (explícitamente o implícitamente). Además, hemos mostrado dos escenarios diferentes donde el protocolo puede ser utilizado ahorrando hasta 3 round trip times. Estos escenarios se sirven de un token de seguridad que permite relacionar la autenticación a la red de acceso y la autenticación a la red IMS. Para obtener este tipo de token criptográfico, se puede utilizar TLS extractor o cualquier otro mecanismo de extracción de material secreto desde un túnel seguro. En consecuencia, este material de claves se puede utilizar para generar un token de seguridad que puede relacionar criptográficamente los dos procesos de autenticación. Para concluir, hemos realizado algunos test en implementaciones comerciales de la pila de protocolo IMS para evaluar la posibilidad de desarrollo del nuevo mecanismo de registro propuesto.

Agradecimientos

Este trabajo ha sido financiado parcialmente por la ayuda a Grupos Investigación CAM CCG08-UC3M/TIC-4479.

Referencias

[1] Puthenkulam, J., et al.: The Compound Authentication Binding Problem. Technical report, IETF (2003)

[2] Asokan, N., Niemi, V., Nyberg, K.: Man-in-the-middle in tunneled authentication protocols. Technical report, In 11th Security Protocols Workshop (2002)

[3] Technical report, 3GPP, Third Generation Partnership Project, Technical Specification Group Services and Systems Aspects, 3G Security, Security Architecture, Technical Specification 3G TS 33.102, V3.7.0 (2000)

[4] Rescorla, E.: Keying material exporters for transport layer security (tls). Technical Report draft-ietf-tls-extractor-05.txt, IETF (2009) http://tools.ietf.org/html/draft-ietf-tls-extractor-05.

[5] Dierks, T., Allen, C.: The transport layer security (TLS) version 1.0. Technical Report RFC2246, IETF (1999) http://www.ietf.org/rfc/rfc2246.txt.

[6] Ohba, Y., Baba, S.: Pana over tls. Technical report, IETF (2002)

[7] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E.: Sip: Session initiation protocol. Technical Report RFC3261, IETF (2002) http://www.ietf.org/rfc/rfc3261.txt.

[8] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., Stewart, L.: Http authentication: Basic and digest access authentication. Technical Report RFC2617, IETF (1999) http://www.ietf.org/rfc/rfc2617.txt.

[9] Peterson, J.: Session initiation protocol (sip) authenticated identity body (aib) format. Technical Report RFC3893, IETF (2004) http://www.ietf.org/rfc/rfc3893.txt.

[10] Veltri, L., Salsano, S., Martiniello, G.: Wireless lan-3g integration: Unified mechanisms for secure authentication based on sip. Communications, 2006. ICC '06. IEEE International Conference on **5** (2006) 2219–2224

[11] Lin, Y.B., Chang, M.F., Hsu, M.T., Wu, L.Y.: One-pass gprs and ims authentication procedure for umts. Selected Areas in Communications, IEEE Journal on **23** (2005) 1233–1239

[12] Celentano, D., Fresa, A., Longer, M., Robustelli, A.: Improved authentication for ims registration in 3g/wlan interworking. Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on (2007) 1–5

[13] Park, J.C., Jun, A.H.: A lightweight ipsec adaptation for small devices in ip-based mobile networks. Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference **1** (2006) 5 pp.–302

La Plataforma Telcoblocks de Despliegue y Desarrollo de Servicios VoIP

Jonathan González
Depto Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid
Ciudad Universitaria s/n 28040 Madrid
jonathan.gsanchez@alumnos.upm.es

Carlos A. Iglesias
División de I+D+i
Germinus XXI (Grupo Gesfor)
Avda Manoteras, 32 28050 Madrid
cif@germinus.com

Felipe Echanique
Depto Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid
Ciudad Universitaria s/n 28040 Madrid
felipe.echanique.torres@alumnos.upm.es

***Resumen*—Este artículo presenta el entorno de desarrollo y despliegue de servicios VoIP propuesto dentro del proyecto Telco-Blocks. En concreto, se detalla el componente de personalización propuesto que facilita la personalización de servicios construidos con tecnologías JAIN SLEE o SIP Servlets. El componente ha sido aplicado al desarrollo de un servicio de personalización de tonos, así como a un servicio de personalización de anuncios en un servicio de Click to Dial.**

***Palabras Clave*—Telcoblocks, personalización, VoIP, Java, SIP, SIP Servlets, JAIN SLEE**

I. Introducción

La convergencia de las redes de telefonía tradicional e Internet [21] está proporcionando una nueva arquitectura para el desarrollo de servicios telco en las así llamadas Redes de Nueva Generación (*Next Generation Networks*; NGN [3]). Una de las principales características de NGNs [2] es el desacoplamiento de redes y servicios, permitiendo que se ofrezca de forma separada y que evolucionen independientemente.

Los principales retos para esta nueva arquitectura NGN son por una parte, proporcionar un entorno de ejecución de servicios tolerante a fallos y con calidad de operadora (*carrier grade*), que sea al menos tan robusto y seguro como la actual Red de Telefonía Conmutada (RTC); por otra parte, debe ofrecer un entorno flexible y abierto que permita desarrollar nuevos servicios de forma ágil, dado que la alta competitividad del entorno telco está demandando esta flexibilidad.

Para conseguir esta flexibilidad, Moyer [21] señala que NGN debe ser más abierta que RTC a los desarrolladores de servicios, ofreciendo APIs abiertas y facilitando bloques de servicios primitivos que los desarrolladores puedan reutilizar. En esta línea, el software de telecomunicaciones NGN ha seguido tres enfoques: JAIN SLEE [16], Parlay [7] y SIP Servlets [22].

El desarrollo de aplicaciones web 2.0 complementa [13] los servidores de aplicaciones híbridos JavaEE / SIP ofrecidos en los modelos de arquitectura orientados a servicios, y cuestiona si la capa de servicios IMS debe seguir siendo una capa separada, orientada a aplicaciones telco, o integrada con las aplicaciones web 2.0. Esta convergencia de servicios telco y web se está comenzando a denominar *mashups telco web2.0* [13]. En esta misma línea, la tecnología web de mashups está comenzando a usarse para la construcción de servicios telco por parte de los usuarios [9].

A pesar de disponer de estas facilidades, el desarrollo de aplicaciones en entornos telco aún no se ha popularizado, debido a la complejidad de las tecnologías involucradas, así como a la falta de entornos que faciliten su adopción. El objetivo del proyecto Telcoblocks [8] es el desarrollo de una plataforma abierta que facilite el desarrollo, despliegue, prueba e integración de servicios VoIP, tanto en operadoras como en empresas que disponen de una centralita VoIP. Telcoblocks es un proyecto orientado a los desarrolladores, con el fin de reducir drásticamente los tiempos de desarrollo y despliegue de nuevos servicios. Uno de los pilares del proyecto es la reutilización de componentes y servicios para la construcción de nuevos servicios.

El resto del artículo se estructura como sigue. La sección II presenta el contexto de esta investigación, que se enmarca en el proyecto TelcoBlocks, detallando la plataforma de ejecución y desarrollo de servicios definida. A continuación, la sección III detalla el componente de personalización, objeto principal de este artículo, que facilita la integración de funcionalidades de personalización en la construcción de un servicio (anuncios, facturación, tarificación, etc.). La sección IV presenta un caso de estudio del componente de personalización presentado en la sección III y desplegado en la infraestructura presentada en la sección II. Por último, se presentan conclusiones del trabajo realizado y las líneas actuales de trabajo en la sección V.

II. Telcoblocks: un entorno para desarrollo, ejecución y prueba de servicios VoIP

Telcoblocks nace con el objetivo de proveer una plataforma abierta para el desarrollo y ejecución de servicios de telecomunicación para la Red de Nueva Generación (NGN[3]) empleando tecnologías JAIN SLEE [16], SIP Servlets [22] y Parlay [7].

El proyecto investiga en el desarrollo de herramientas de software libre para entornos telco que cumplan con los fuertes requisitos de disponibilidad y fiabilidad de estas redes. Como resultado, la plataforma resultante se liberará como código abierto con licencia GPL, lo cual supone un cambio de mentalidad frente el uso exclusivo de software propietario en este tipo de entornos.

El proyecto también innova en la aplicación de sistemas inteligentes a las telecomunicaciones. Frente el fenómeno emergente de la sobre-comunicación potenciado por el auge de las comunicaciones personales, el proyecto investiga en la integración de técnicas inteligentes para gestionar dichas comunicaciones según las preferencias de los usuarios, así como para facilitar la personalización de servicios.

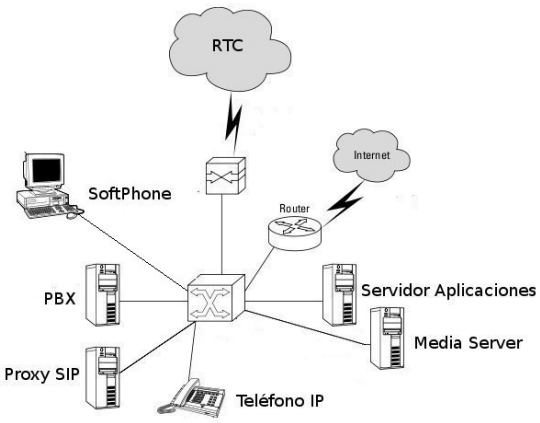


Figura 1. Plataforma de Despliegue de Servicios VoIP de Telcoblocks

Telcoblocks enmarca los aspectos mencionados dentro de procesos de desarrollo ágil de servicios telco basados en tecnologías abiertas y estándares, cuyo ahorro en costes y menor time-to-market abre interesantes oportunidades de negocio a las PYMES para proporcionar servicios de telecomunicación a usuarios finales.

Telcoblocks desarrolla dos plataformas: la plataforma para despliegue de servicios (sección II-A) y la plataforma de desarrollo de servicios (sección II-B).

II-A. Plataforma de Despliegue de Servicios VoIP

En los últimos años, la gran aceptación que ha tenido la VoIP como solución a las comunicaciones corporativas de las empresas ha favorecido al nacimiento de algunos proyectos de software abierto que permiten implementar la mayoría de las características que ofrecen algunas de las PBX comerciales (Avaya, Cisco...) que desde el nacimiento de esta tecnología han sido utilizadas por las grandes y medianas empresas.

En la actualidad existen diferentes soluciones de software abierto para del despliegue de una infraestructura VoIP basada en el protocolo SIP [18]. TelcoBlocks pretende investigar en la integración de estas soluciones para crear una arquitectura integrada que permita generar un entorno de pruebas y despliegue de servicios de forma sencilla utilizando diferentes elementos.

El entorno básico de despliegue de servicios VoIP de Telcoblocks se ilustra en la figura 1 y consta de los siguientes elementos, que serán descritos en la siguiente sección:

- Centralita VoIP (PBX) (sección II-A1).
- Proxy SIP - Servidor de Registro (sección II-A2).
- Servidor Multimedia (sección II-A3).
- Servidor Telco de Aplicaciones (sección II-A4).

II-A1. Centralita VoIP (PBX): Una PBX (*Private Branch Exchange*) es una central telefónica conectada directamente a la red pública de teléfono por medio de líneas troncales para gestionar, además de las llamadas internas, las entrantes y/o salientes con autonomía sobre cualquier otra central telefónica. Los usuarios de una PBX no tienen asociada ninguna central de teléfono pública, ya que es el mismo PBX que actúa como tal, análogo a una central pública que da cobertura a todo un sector mientras que un PBX lo ofrece a las instalaciones de una compañía generalmente.

Una centralita VoIP (PBX) ofrece:

- Salida a la Red Telefónica Conmutada (RTC): Los usuarios registrados en el sistema podrán realizar llamadas a la RTC.
- Llamadas a bajo coste: Una PBX es capaz de conectarse a diferentes proveedores de servicios de VoIP, permitiendo así la realización de llamadas que serán tarificadas a la plataforma por el proveedor con el que se realice la llamada.

La solución de código abierto más empleada en el ámbito profesional es Asterisk [5]. Asterisk es una aplicación de software libre que implementa las funciones de una central telefónica (PBX). La aplicación es compatible con diversos sistemas operativos (Linux, MacOS, Solaris y Microsoft Windows) aunque es mejor soportada en sistemas Linux.

Como toda PBX, Asterisk permite la interconexión de diferentes teléfonos que pueden ser registrados en la aplicación y ser localizados entre sí mediante la marcación de diferentes extensiones definidas en un plan de llamadas. Además, permite conectarse con proveedores de llamadas VoIP de forma que un teléfono registrado en Asterisk es capaz de realizar llamadas a números de la RTC a bajo precio.

Una de las características más importantes de Asterisk es el hecho de que puede ser usado con diferentes protocolos de señalización de VoIP; entre ellos destacar H.323 y SIP aunque también es muy utilizado el protocolo IAX (propietario de Asterisk) [19] para comunicación entre diferentes PBX de este tipo.

El hecho de ser una aplicación de software libre la hace realmente atractiva, debido a las mejoras que se van añadiendo en las nuevas versiones que son publicadas cada poco tiempo. Actualmente se encuentra publicada la versión 1.4.x de Asterisk a la espera de que la versión de 1.6.x sea publicada en los próximos meses.

Un proyecto que ha sido analizado por su sencillez de configuración es Elastix [27] que incluye junto con un Asterisk configurable desde una interfaz web (FreePBX) un sistema de facturación muy sencillo de configurar.

II-A2. Proxy SIP - Servidor de Registro:

Cada usuario tiene una dirección lógica que es invariable respecto de la ubicación física del usuario. Una dirección lógica del protocolo SIP es de la forma usuario@dominio es decir tiene la misma forma que una dirección de correo electrónico. La dirección física (denominada ”dirección de contacto”) es dependiente del lugar en donde el usuario está conectado (de su dirección IP).

Cuando un usuario inicializa su terminal (por ejemplo conectando su teléfono o softphone SIP), el agente de usuario SIP que reside en dicho terminal envía una petición con el método REGISTER a un Servidor de Registro, informando a qué dirección física debe asociarse la dirección lógica del usuario. El servidor de registro realiza entonces dicha asociación. Esta asociación tiene un período de vigencia y si no es renovada, caduca. También puede terminarse mediante un desregistro. La forma en que dicha asociación es almacenada en la red no es determinada por el protocolo SIP, pero es vital

que los elementos de la red SIP accedan a dicha información.

Además, en el caso de la arquitectura propuesta, este servidor actúa también como servidor proxy o de redirección. Para encaminar un mensaje entre un agente de usuario cliente y un agente de usuario servidor normalmente se recurre a estos servidores. Estos servidores pueden actuar de dos maneras:

1. Como Proxy, encaminando el mensaje hacia el destino.
2. Como servidor de Redirección generando una respuesta que indica al originante la dirección del destino o de otro servidor que lo acerque al destino. La principal diferencia es que el servidor proxy queda formando parte del camino entre los extremos de la comunicación, mientras que el servidor de redirección una vez que indica al llamante cómo encaminar el mensaje ya no interviene más.

En el marco del proyecto Telcoblocks, se han estudiado las posibles soluciones de código abierto que se podían emplear en esta arquitectura entre las que destacan: SER (SIP Express Router) [4], OpenSER [6] (OpenSIPS, Kamailio). Tras un análisis de cada uno de los proyectos se optó por el uso de OpenSIPS, ya que parte de la última versión de OpenSERadeacuadamente y es de las soluciones más utilizadas en el ámbito profesional junto con Asterisk.

OpenSIPS es una aplicación que implementa un servidor SIP. Originalmente pertenecía a otro proyecto que sigue estando activo llamado SER (*Sip Express Router*). La principal diferencia entre ambos proyectos es el hecho de que OpenSER está llevado por una comunidad en lugar de una empresa, lo que hace que rápidamente se implementen mejoras en el servidor y las empresas se inclinen por la opción de uso de OpenSER. OpenSIPS es muy versátil para su implantación, permitiendo su instalación en sistemas con recursos limitados así como en grandes servidores. Está escrito completamente en C y orientado principalmente a equipos Linux/Unix. Esta aplicación tiene muchas características interesantes, entre las que podemos destacar las siguientes: Location Service, registrar, servidor Proxy, servidor de redirección de llamadas, gateway hacia otras redes que no son SIP. Al igual que Asterisk, OpenSIPS permite la interconexión con diferentes terminales y llamadas a través de la RTC.

II-A3. Servidor Multimedia: Un Servidor Multimedia (*Media Server*) permite la implementación de muchos de los servicios mencionados en la descripción de la centralita VoIP, la diferencia fundamental es que no registra usuarios sino que solo se encarga del tráfico RTP.

De entre las soluciones que implementan un servidor multimedia, han sido consideradas la solución de Mobicents (*Mobicents Media Server* [28]) y la solución del proyecto SER, llamada SEMS [26]. Por facilidad de integración con el servidor SIP escogido se ha optado por esta última opción.

SEMS es un servidor de aplicaciones y recursos multimedia para servicios VoIP basados en SIP. Presenta muy buenas prestaciones realizando algunos servicios básicos como reproducción de anuncios o servicio de conferencia combinándose, en algunos casos, con servidores de aplicaciones externos. Gracias a su facilidad de uso y su framework de aplicaciones flexible se puede unir en un mismo proceso la lógica de la aplicación con los recursos servidos por el servidor. SEMS suele ser empleado con algunos de los servidores SIP analizados anteriormente (SER, OpenSER, OpenSIPS) de tal manera que se obtiene así un servicio de VoIP completo. Las principales funcionalidades de SEMS son las siguientes:

- Servicio de Conferencia: Permite conversaciones telefónicas por más de dos usuarios simultáneamente.
- Servicio de Videoconferencia.
- Anuncio: Reproduce un recurso solicitado.
- Servicio de Voicemail: Servicio que almacena mensajes destinados a usuarios que en el momento de la recepción de los mismos no pudieron atender la llamada. Estos mensajes son enviados posteriormente al correo electrónico del usuario.

II-A4. Servidor Telco de Aplicaciones: Ejecuta la lógica de negocio del servicio (ClickToDial, tarificación de llamadas, personalización de servicios) Telcoblocks actualmente soporta dos tecnologías: JAIN SLEE [16] y SIP Servlets [22]. Como servidor JAIN SLEE, se ha escogido Mobicents [24], que soporta JAIN SLEE y SIP Servlets. Como servidor de SIP Servlets se ha escogido Sailfin [1], que soporta SIP Servlets, gracias a sus facilidades de desarrollo integradas con el IDE Netbeans.

Una vez analizadas cada una de las soluciones escogidas para la arquitectura se puede concluir afirmando que las principales ventajas de esta arquitectura son:

1. Escalabilidad de servicios, gracias al uso de un Servidor Telco de aplicaciones.
2. Soporte al protocolo AAA [25]: Integración Radius [12], Diameter [23], LDAP [20], gracias a la integración de OpenSIPS.
3. Facturación y conectividad a RTC ofrecidas por Asterisk.

La plataforma de despliegue se ha empaquetado como una **máquina virtual** con VmWare [29], para que los desarrolladores puedan disponer de un entorno donde puedan probar rápidamente los servicios desarrollados. Dicha máquina virtual, contiene todos los elementos mencionados en la arquitectura, adecuadamente configurados, para que el desarrollador sólo deba estar pendiente del nuevo servicio que esté implementando.

II-B. Plataforma de Desarrollo de Servicios VoIP

Actualmente, el desarrollo de servicios telco manifiesta ciertas deficiencias, como son la baja productividad, el elevado tiempo dedicado al desarrollo así como las altas habilidades que necesitan los desarrolladores para conocer e integrar los actuales frameworks y componentes.

Para resover estas deficiencias, Telcoblocks pretende abordar dos direcciones: mejorar la prueba de los servicios, y reducir el tiempo de desarrollo mediante la reutilización de componentes, que redundarán en una mejora de su calidad.

La plataforma de desarrollo requiere que se facilite el desarrollo con dos tecnologías, JAIN SLEE y SIP Servlets, que se presentan a continuación brevemente.

II-B1. JAIN SLEE: JAIN SLEE [16] es un modelo de componentes definido para un servidor de aplicaciones diseñado específicamente para aplicaciones telco. Está concebido como una plataforma de procesamiento de eventos de altas prestaciones y tolerante a fallos. Frente a las aplicaciones de empresa y web, síncronas e intensivas en datos por naturaleza,

y que pueden ser modeladas e implementadas de forma adecuada con la tecnología JEE (*Java Enterprise Edition*), SLEE está dirigido a aplicaciones asíncronas, como son las aplicaciones telco, y a procesar eventos de red combinando múltiples protocolos. En la actualidad JAIN SLEE es el estándar Java para entornos de ejecución de lógicas de servicio (*Service Logic Execution Environment*) [30].

La principal ventaja de JAIN SLEE es la estandarización del desarrollo y ejecución de servicios, de modo que se cubran los diferentes niveles y, muy especialmente, la capa de acceso a los elementos de telecomunicaciones. A pesar de la estandarización en el acceso a las capacidades de red ofrecida por protocolos como INAP o CAP, los cuales permiten la interconexión de redes, la complejidad de la implementación de servicios de forma directa y los requerimientos de las telcos de implementación (rendimiento, distribución, confiabilidad, etc.) hacen del desarrollo de servicios una actividad muy compleja, que cuenta con tan sólo un reducido grupo de profesionales con la competencia adecuada. JAIN SLEE, por el contrario, ofrece un alto nivel de abstracción para el acceso a las capacidades de red, simplificando en gran medida la complejidad existente y, al mismo tiempo, ofreciendo Java como lenguaje de implementación. Paralelamente, JAIN SLEE incorpora los conceptos tradicionales de la arquitectura Java (reutilización de componentes, facilidades de administración y concurrencia, distribución, etc.).

JAIN SLEE, como paradigma de los entornos de nueva generación, rompe con la tradicional distinción entre servicios de inteligencia de red y servicios IP. El concepto de adaptadores de recursos (RA) permite la integración entre diferentes tecnologías en servicios innovadores y de nueva generación, así como el despliegue de servicios tradicionales basados en señalización, como la gestión de enrutado de llamadas. No existe un único campo de servicios adecuado para JAIN SLEE; por el contrario, una de las principales ventajas es su gran alcance, el cual es además ampliable mediante la incorporación de adaptadores de nuevos protocolos. Otra de las ventajas es su definición abierta, que elimina las barreras tradicionales de integraciones propietarias.

II-B2. SIP Servlets: SIP Servlets [22] proporcionan un modelo de desarrollo específico del protocolo SIP, dado su papel fundamental en las nuevas arquitecturas IMS. Es una tecnología alternativa a JAIN SLEE, que ofrece un modelo de desarrollo más sencillo, aunque no permite manejar otros protocolos diferentes de SIP y no es transaccional.

Telcoblocks va a contemplar una línea de investigación basada en el uso de una biblioteca de componentes, el cual ayudará a los ingenieros software a la selección e integración de bloques para la creación de nuevos servicios.

La plataforma de desarrollo de telcoblocks, está formada por los siguientes elementos:

- *Plugins para Eclipse y Netbeans*, que facilitan el desarrollo de forma gráfica de nuevos servicios. Actualmente se ha implementado un diagrama de despliegue para definir y configurar la plataforma de despliegue de servicios, y un diagrama de bloques de servicio, para combinar módulos desarrollados previamente.
- *Herramientas de prueba y simulación*. El proyecto integrará el desarrollo de pruebas automáticas así como la integración de pruebas que faciliten el desarrollo de servicios.
- *Biblioteca de componentes*. A través del desarrollo de casos de uso, se han identificado algunos componentes básicos que se reutilizan en numerosos servicios. El primero que se ha implementado es el componente de personalización, que se presenta a continuación.

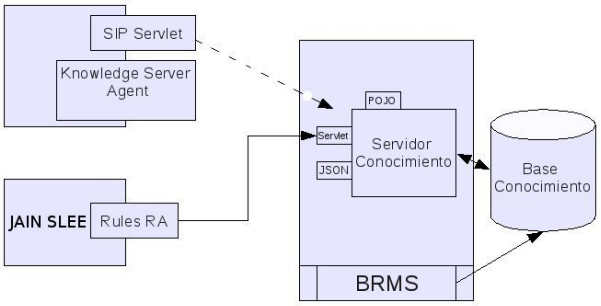


Figura 2. Arquitectura del Componente de Personalización

III. Componente de Personalización

El objetivo del componente de personalización es facilitar la externalización de la lógica del negocio de un servicio de telecomunicaciones. La personalización [17] es un elemento clave para tanto el descubrimiento de nuevos servicios, como la adaptación de los servicios existentes a las características de los usuarios.

La figura 2 muestra la arquitectura del componente de personalización propuesto, que consta de los siguientes elementos:

- **Servidor de Conocimiento:** es el encargado de gestionar la *base de conocimiento*, ofreciendo una interfaz para su acceso remoto La base de conocimiento define las reglas y hechos de la lógica de servicio. El servidor está integrado con otro componente, llamado BRMS (*Business Rule Management System*) que permite la edición de la base de conocimiento mediante una interfaz web. Se ha seleccionado la plataforma Drools [11] para su implementación. Ofrece diferentes interfaces para su acceso (clase Java en local, útil para desarrollo, Servlet para acceso remoto, o bien JSON para las capa de presentación).Para el acceso remoto de este servidor de conocimiento, los clientes emplean *RuleAgents*, que son clases Java que hacen de proxy del servidor de conocimiento, encapsulado toda la lógica de acceso.
- **Interfaz para tecnología SIP Servlets**. Los SIP servlets atienden las peticiones SIP, haciendo peticiones al servidor de conocimiento cuando de él precisa algo, por ejemplo; en el servicio de anuncio personalizado, realiza una petición al servidor de conocimiento, para saber el anuncio a reproducir, antes de solicitar el anuncio a reproducir al servidor multimedia. Los SIP Servlets usan los componentes *RuleAgent* directamente.
- **Interfaz para tecnología JAIN SLEE**. Para los componentes JAIN SLEE, la integración se ha realizado mediante un adaptador de recursos Rules-RA [10], que permite cargar realizar consultas a los componentes

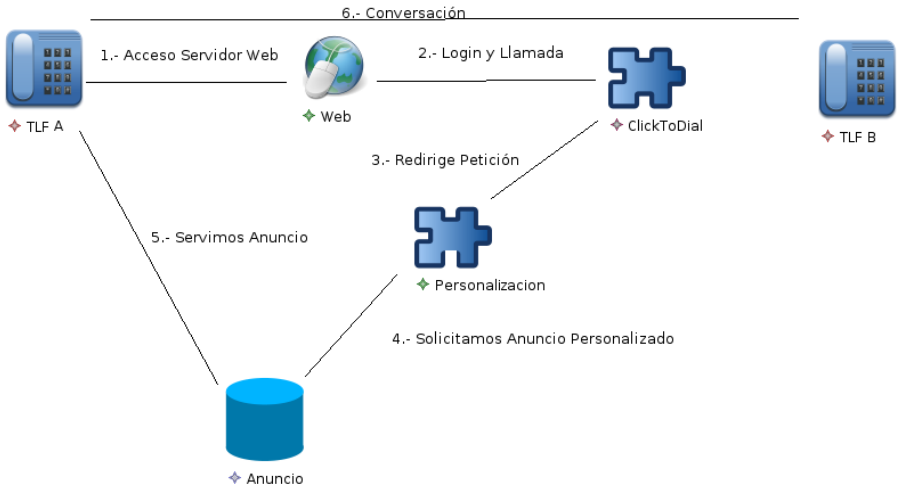


Figura 3. Secuencia Acciones Servicio

Figura 4. Interfaz Web Click To Dial

SBB. En este caso, es el adaptador de recurso el que emplea un componente *RuleAgent* para acceder al gestor de conocimiento, que es configurable mediante la interfaz de gestión con JMX.

IV. Caso de estudio: Personalización de Anuncios para el llamante

El servicio seleccionado consiste en la personalización de anuncios a un llamante combinado con un servicio de llamadas a través de la web (*clickToDial*, figura 4). Los usuarios se registran en un portal web, especifican una serie de datos en su perfil, y pueden realizar llamadas pinchando en un botón (servicio *ClickToDial*). En el momento en el que se cursa la llamada, el sistema selecciona un anuncio auditivo, que es escuchado por el usuario, y una vez que termina el anuncio se cursa la llamada al destinatario. A cambio de escuchar la llamada, el usuario puede recibir alguna promoción, tales como descuentos o minutos gratis.

La figura 3 muestra el diagrama de servicio desarrollado con el plugin para diagramas de bloques de Telcoblocks para Eclipse, desarrollado con *Graphical Modeling Framework* (GMF) [15]

Primeramente, intentaremos abordar el problema desde el punto de vista de SIP. En el escenario de este servicio debe haber, al menos, dos clientes SIP registrados en nuestro servidor (llamante y destinatario de la llamada). El intercambio completo de mensajes durante la reproducción de anuncio y durante la conversación se muestra en la figura 5.

Una vez registrados ambos terminales, se procede a la realización de la llamada. La petición SIP de llamada(INVITE) es dirigida al registrar, que va a dirigir dicha petición al servidor multimedia (SEMS [26]) que reproducirá el anuncio solicitado. La cabecera de la petición enviada por el registrar al media server tiene un formato especificado en la RFC 4240 [14]. En la cabecera de esta petición debe ir especificado el anuncio que debe ser reproducido por el servidor multimedia, en un parámetro llamado *"play="*. Una vez finalizada la reproducción del anuncio, se finaliza la sesión para negociar la sesión con el destinatario de la llamada.

Para realizar la selección del anuncio, se emplea el componente de personalización previamente descrito. La base de conocimiento consta de tres inferencias como se muestra en el diagrama de inferencias (figura 6: segmentación de la población, clasificación de anuncios y selección del anuncio. Tal como se indica, actualmente las dos primeras inferencias se realizan en la fase de registro, mientras que la selección del anuncio se realiza en tiempo real con cada llamada.

La segmentación de la población, consiste en clasificar cada uno de los usuarios de la plataforma en cada uno de los segmentos de población que se han definido a partir de criterios de edad y estudio/trabajo, considerando este último criterio fundamental para determinar el nivel social del usuario (poder adquisitivo). Esta segmentación se realizará cada vez que un usuario se registra en el sistema, o modifica uno de los campos que afecta a la segmentación. No es necesario hacer una segmentación de la población entera cada vez que se modifica un usuario, sino que basta con solo insertar a ese usuario en la base de conocimiento, partiendo del hecho de que los perfiles de usuario son estáticos en este caso.

La clasificación de los anuncios sigue criterios parecidos, clasificándolos en categorías de contenidos y precios cuando los anuncios son dados de alta o modificados. Asi pues la

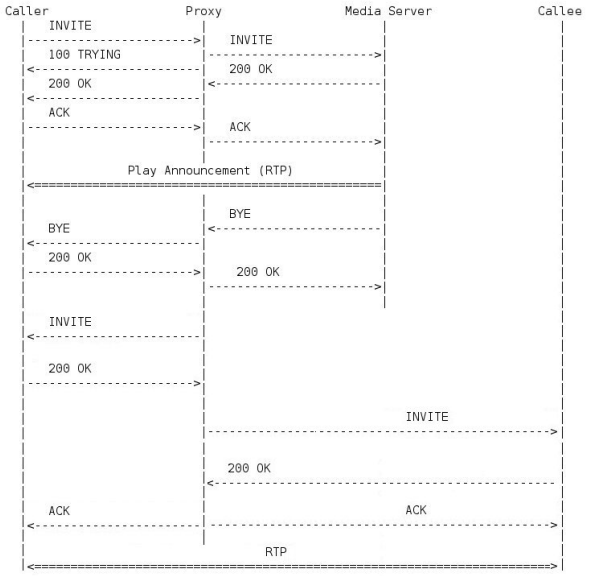


Figura 5. Diagrama de Trazas SIP en el servicio de anuncio personalizado

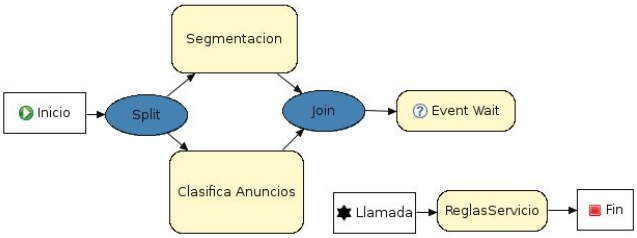


Figura 6. Flujo de Inferencias para la personalización de un anuncio

clasificación de los anuncios atiende a los siguientes criterios:

- Edad, diferenciando si van dirigidos a mayores de edad o no.
- Temática (Ocio, Cultura, Evento...)
- Precio, distinguiendo si va dirigido a un público con un poder adquisitivo elevado o no.

La última fase es la fase de elección del anuncio. Se emplea una técnica de puntuación (*scoring*) de los anuncios según su afinidad al perfil del llamante (por edad, por aficiones, por nivel social (trabajo), región, etc) y a algunas características que tengan en común llamante, usuario llamado y anuncio. Los usuarios y anuncios ya han sido procesados previamente, por lo que a partir de los datos del usuario que realiza la llamada, se puede inferir la temática del anuncio más adecuada para dicho usuario. Así pues, los anuncios de dicha temática son puntuados según su afinidad con el usuario que realiza la llamada escogiéndose finalmente el que mayor puntuación obtenga. Este anuncio será reproducido por el servidor multimedia dejando paso, una vez finalizada la reproducción del citado anuncio, a la conversación con el usuario seleccionado previamente.

V. Conclusiones y Futuros Trabajos

El artículo ha presentado las plataformas de despliegue y desarrollo para servicios VoIP de Telcoblocks, que facilitan y reducen el tiempo de desarrollo de nuevos servicios VoIP.

El componente de personalización facilita su reutilización en varios niveles. Por una parte, facilita la inclusión de personalización en nuevos servicios, reutilizando los componentes desarrollados y simplemente definiendo una nueva base de conocimiento. Por otra parte, la misma base de conocimiento puede ser usada por varios servicios programados con tecnologías diferentes, gracias al uso de un servidor de conocimiento.

El caso de estudio presentado ha mostrado la facilidad para integrar, por ejemplo, la RFC 4240 en el desarrollo del servicio de anuncios personalizados.

Actualmente, se está trabajando en el uso del componente de personalización en otros ámbitos. Se ha desarrollado su integración con un GoogleTalk mediante componentes JAIN SLEE, lo que muestra la flexibilidad del componente desarrollado.

Otra línea de trabajo actual es el entorno de desarrollo de Telcoblocks, para el que se están desarrollado plugins para Eclipse, realizados con EPF, como se han mostrado en este artículo.

Agradecimientos

Este proyecto ha sido cofinanciado por el Ministerio de Industria, Turismo y Comercio mediante el proyecto Telco-Blocks (TSI-020302-2008-16) bajo el programa Avanza I+D 2008.

Referencias

[1] Sailfin - contenedor de sip servlets. Technical report, sun.
[2] Conclusions from the ngn-sg. Technical report, ETSI 38th General Assembly meeting Nice, November 2001.
[3] Definition of next generation network. Technical report, ITU, 2004.
[4] Sip express router (ser. Technical report, iptel.org, 2004.
[5] Asterisk the future of telephony. Technical report, asterisk.org, 2005.
[6] Open sip express router (ser). Technical report, openser.org, 2007.
[7] Osa parlay x 3.0 specifications. Technical report, ETSI, 2007.
[8] Proyecto telcoblocks, 2008. Disponible en http://telcoblocks.germinus.com.
[9] A. Alvaro Martínez Reol, C. Baladrón Zorita, A. León Martín, C. García Morchón, L. Calavia Domínguez, J. Aguiar Pérez, and J. Caetano. Nuevos modelos de negocio: Servicios generados por el usuario. In *XVIII Jornadas Telecom I+D, ISBN-13: 978-84-9860-135-0, Octubre 2008, Bilbao*, 2008.
[10] A. Bhayani. Mobicents rules-ra, disponible en http://groups.google.com/group/mobicents-public/web/mobicents-rules-ra, 2007.
[11] P. Browne. *JBoss Drools Business Rules*. PACKT Publishing, 2009.
[12] A. R. C. Rigney, S. Willens. Remote authentication dial in user service (radius). Technical report, IETF - RFC 2865, 2000.
[13] C. Chappell. IMS's web 2.0 problem. *Light Reading's Services Software Insider*, 2007.
[14] A. S. E. Burger, J. Van Dyke. Basic network media services with sip. Technical report, ietc - RFC 4240, 2005.
[15] Eclipse. Graphical modeling framework tutorial, disponible en http://wiki.eclipse.org/index.php/gmf_tutorial. Technical report, 2006.
[16] D. Ferry and S. Lim. JSR 22. JAIN SLEE API Specification. Technical report, Java Community Process, Mar. 2004.
[17] R. Guarneri, A. M. Sollund, D. Marston, E. Fossbak, B. Berntsen, G. Nygreen, G. Gylterud, R. Bars, and A. Kerdraon. Report on the state of the art in personalisation, common framework. Technical report, ePerSpace - IST Integrated Project, 2004.
[18] G. J.Rosenberg, H.Schulzrinne. Sip: Session initiation protocol. Technical report, ietf, 2002.
[19] F. M. M. Spencer, B. Capouch. Iax: Inter-asterisk exchange version. Technical report, ietf - RFC 5456, 2009.

[20] S. K. M. Wahl, T. Howes. Lightweight directory access protocol. Technical report, IETF - RFC 2251, 1997.
[21] A. Moyer, S.Umar. The impact of network convergence on telecommunications software. *Communications Magazine*, 2001.
[22] P. O'Doherty. Jsr 289 - sip servlet v1.1. Technical report, Java Community Process, 2008.
[23] E. G. P. Calhoun, J. Loughney. Diameter base protocol. Technical report, IETF - RFC 3588, 2003.
[24] L. Red HatMiddleware. Mobicents. the open source slee and sip server. disponbile en http://www.mobicents.org/index.html, 2009.
[25] P. C. S. Farrell, J. Vollbrecht. Aaa authorization requirements. Technical report, ietf - RFC 2906, 2000.
[26] S. Sayer. Sems-ng design overview inside the media server. Technical report, iptego GmbH, 2006.
[27] B. Sharif. Elastix without tears. Technical report, PaloSanto Solutions, 2008.
[28] D. Silas. Mobicents media server guide. Technical report, JBoss, 2008.
[29] VMware Inc. Página web de VMware, disponible en http://www.vmware.com, 2009.
[30] Ángel Cruz. Una nueva convergencia: ¿java en la red? Technical report, 2005.

Evaluación de Prestaciones de RSVP Extendido para un Escenario con Garantías de Distribución

Marcos Postigo Boix y José L. Melús Moreno
Departamento de Ingeniería Telemática
Universidad Politécnica de Cataluña …
C. Jordi Girona 1-3. Campus Nord. 08034 Barcelona.
marcos.postigo@entel.upc.edu, teljmm@entel.upc.edu

***Resumen*- In some guaranteed data delivery scenarios, servers can quickly and carefully allocate the available bandwidth among the requests of the users to reduce rejections. This is particularly important when reservations are higher than minimum required delivery rate (semi-elastic reservations). As an example, when semi-elastic reservations utilize all the available server bandwidth and a new flow reservation arrives, it is useful to reallocate current reservations to accept the new one. The RSVP protocol is receiver oriented and it is in charge of setting up these reservations. However, in some cases, to reallocate bandwidth in a receiver oriented way could delay the required sender reservation adjustments. This paper presents a new extension of the RSVP signaling messages that allows the server to adjust and modify these reservations. The performance evaluation of the extended and the native RSVP signaling protocols when used to manage the access bandwidth of a semi elastic flows server shows the benefits of the extensions. In particular, the use of resource reservation is reduced because of a more efficient bandwidth usage. In addition, the blocking probability is also reduced because of a more flexible bandwidth reallocation situation. The authors believe that this procedure is an intermediate step to improve the lack of flexibility that RSVP presents.**

***Palabras Clave*- Sender-initiated resource reservation, RSVP, signaling protocol messages, semi-elastic flows.**

I. INTRODUCCIÓN

El IETF escogió como estándar el protocolo "*resource ReSerVation Protocol*" (RSVP) [1][2][3][4]. Sin embargo, la especificación de RSVP tiene problemas significantes en muchos contextos. Otras contribuciones de diferentes autores proponen extensiones a RSVP para solucionar esas carencias. En [5], los autores presentan extensiones experimentales de RSVP para el empleo de clientes remotos que tienen acceso a un demonio RSVP y el empleo de reservas en una pasada. La referencia [6] presenta una versión ligera de RSVP que elimina la capacidad multicast y separa claramente los datos de los mensajes de señalización. La referencia [7] define una nueva extensión de RSVP llamada "*On Board RSVP protocol*" que apoya eficazmente la reserva de recursos extremo a extremo para la movilidad de los routers.

En [8] los autores proponen nuevas extensiones para RSVP que proporcionan servicios en tiempo real en entornos IPv6 móviles y jerárquicos.

En [9] los autores muestran un esquema de reserva de recursos dinámico para reducir la interrupción del servicio de aplicaciones en tiempo real debido a la frecuente movilidad de los terminales en las redes inalámbricas. La referencia [10] presenta una extensión de RSVP para proporcionar un "*Dynamic RSVP*" (DRSVP). Esta aproximación ajusta dinámicamente los recursos reservados en los nodos sin demasiado esfuerzo. Además, en [11] los autores extienden DRSVP para soportar "*soft-handoffs*" en redes IP móviles e inalámbricas. En [12] los autores también proponen un protocolo de reservas de recursos con soporte móvil (Mobile RSVP) que crea un camino móvil que se puede adaptar a la topología de encaminamiento según el movimiento de los nodos.

En [13] se analiza una extensión de RSVP para garantizar la calidad de servicio de las aplicaciones móviles de Internet. Además, se propone un algoritmo para realizar un mejor uso de los recursos de la red.

En [14], los autores proponen nuevas extensiones de RSVP en MPLS. En [15] "*Security Service RSVP*" extiende RSVP para proporcionar un mecanismo que dinámicamente negocia la QoS entre emisores y receptores de las aplicaciones multicast en Internet. La referencia [16] define una extensión de ingeniería de tráfico para RSVP conocida como RSVP-TE para un dominio administrativo y los mensajes de señalización siguen un camino de conmutación de etiquetas MPLS.

En este artículo, nos centramos en la extensión de RSVP que ayuda a un emisor a reorganizar las reservas de forma adecuada cuando su ancho de banda para reservas está totalmente reservado y llegan nuevas reservas al servidor. El principal problema de RSVP en este contexto es que las reservas se inician en el receptor. En algunos escenarios unicast, especialmente cuando el emisor necesita renegociar las reservas de recursos (ReR), es importante ofrecer al servidor la posibilidad de ajustar las reservas directamente, sin la intervención del receptor. Esto puede reducir notablemente el retardo de renegociación. Los flujos semi-elásticos, a diferencia de los flujos inelásticos, no tienen requerimientos estrictos de QoS y pueden permitir la reducción de los recursos reservados e incluso la liberación de dichos recursos sin cambiar sus niveles de QoS esperados. Este es el caso, por ejemplo, de la transmisión de video pregrabado, que puede transmitirse a una tasa mayor que la

tasa de lectura del cliente. Si en algún momento de la transmisión la tasa de envío debe reducirse, el cliente puede continuar leyendo los datos que previamente ha almacenado en su memoria.

Para los flujos semi-elásticos es esencial una gestión correcta de las reservas cuando el ancho de banda disponible cambia dinámicamente. En [17], los autores definen escenarios donde la disponibilidad de ancho de banda es muy variable, como en las redes inalámbricas, y la reserva de ancho de banda se puede beneficiar de las renegociaciones. Consecuentemente, los gestores del ancho de banda pueden elaborar esquemas que maximicen el beneficio para todas las conexiones basándose en la elasticidad de los flujos de datos.

La referencia [18] presenta una solución que usa la planificación y el control de admisión para mejorar las prestaciones para usuarios Premium. El servidor clasifica las peticiones Premium como de alta prioridad y les da un trato preferente. En [19] los autores proponen varios métodos para incrementar la cantidad máxima de clientes que un servidor puede aceptar degradando la calidad de los videos distribuidos. En [20] y [21] los autores proponen acciones similares para distintos escenarios y aplicaciones. En [22] se comparan diferentes mecanismos para distribuir los mensajes de control RSVP y en [23] los autores estudian las interacciones entre los parámetros de temporización de RSVP y las métricas de prestaciones de red como un problema multi-objetivo.

En 2001, el IETF formó un nuevo grupo de trabajo, el *Next Steps In Signaling* (NSIS). Como resultado se desarrolló una nueva arquitectura extensible de señalización IP de dos capas [24]. En [25], los autores discuten protocolos aplicaciones de los protocolos, en particular, se resalta el protocolo de señalización de calidad de servicio (QoS). Las referencias [26][27][28] del NSIS *Working Group* definen los protocolos de señalización iniciados por el emisor como parte del marco de trabajo de NSIS. En [29] presentamos un método para distribuir flujos de información prealmacenada a varios clientes. Usando ReR (Reservas de Recursos), el emisor envía datos en unicast de forma más rápida de lo que los clientes requieren. Cuando la memoria de los clientes alcanza un umbral, no es necesario mantener el modo ReR y cambia a Best-Effort (BE) reduciendo el coste de transmisión e incrementando la eficiencia de las reservas de recursos. De forma similar, el cliente solicita el modo de transmisión ReR, si los datos alcanzan un umbral mínimo. En este escenario multicliente-servidor es esencial controlar el ancho de banda de acceso reservado, ya que un nueva reserva puede ser rechazada si todos los recursos disponibles están reservados.

En este artículo, describiremos la inclusión de nuevos mensajes RSVP y su procesamiento para prevenir rechazos de reservas de recursos en el servidor. Estos nuevos mensajes permiten la modificación de las reservas previas antes de que se acepte una nueva reserva. Estas modificaciones de RSVP pueden considerarse como un paso intermedio para mejorar la falta de flexibilidad de RSVP. Este es un paso previo para obtener la flexibilidad que NSIS ofrecerá cuando esté ampliamente implementado [27][28].

El resto del artículo se organiza de la siguiente forma: la sección 2 define los nuevos mensajes RSVP; la sección 3 describe como el demonio RSVP procesa los mensajes y muestra su uso en escenario homogéneo con varios clientes y un servidor de flujos semi-elásticos; la sección 4 compara la evaluación del protocolo RSVP nativo con el RSVP extendido; finalmente, la sección 5 concluye el artículo.

II. Nuevas Extensiones RSVP y su Implementación

Cuando se usa el protocolo RSVP, los receptores se encargan de establecer y mantener las reservas. De esta forma, si el emisor (servidor) necesita modificar o liberar la reserva, no puede hacerlo directamente. Así, el emisor debe indicar cualquier cambio al receptor que finalmente será quien cambie la reserva.

En un escenario multicliente-servidor que utilice conexiones unicast con cada cliente, cuando los clientes solicitan simultáneamente flujos semi-elásticos, el servidor debe controlar y gestionar adecuadamente el ancho de banda de acceso utilizado por las reservas, para maximizar el número de flujos admitidos. Dado que el servidor es el único que conoce exactamente cuántas reservas están accediendo al servidor, debe ser el responsable de gestionar esos flujos simultáneos. Para garantizar esta funcionalidad, proponemos extender RSVP para permitir al servidor el modificar directamente las reservas. En esta sección, primero definiremos los mensajes RSVP nuevos necesarios y después describiremos la implementación de estas extensiones.

A. Nuevas Extensiones RSVP

Para asegurar al servidor el redistribuir el ancho de banda entre las reservas actuales, definimos tres nuevos mensajes: RESV_S, RESV_S_TEAR y RESV_S_ERROR. Viajan respectivamente en la dirección opuesta a los mensajes RESV, RESV_TEAR y RESV_ERR en RSVP. El mensaje RESV_S se encarga de ajustar las reservas y actualizar los valores de reserva de los mensajes RESV de refresco (Fig. 1). RESV_S_TEAR cancela las reservas y borra el estado específico de reserva. RESV_S_ERROR señaliza estados de error cuando se intenta cambiar una reserva con mensajes RESV_S. Avisa de un error cuando se procesa un mensaje RESV_S y se encamina salto-a-salto usando el estado de path, de forma parecida a los mensajes PATH_ERR.

Para informar de la llegada de los mensajes RESV_S y RESV_S_TEAR al receptor, se define la llamada RESV_S_EVENT. Esta se usa en el receptor para notificar a la aplicación local que el servidor ha cambiado la reserva. Otra llamada (RESV_CONTROL_EVENT) se utiliza para prevenir al emisor (servidor) de un rechazo de reserva, permitiendo a la aplicación la redistribución de ancho de banda antes de comenzar el procesado del mensaje RESV entrante (Fig. 1). Así, cuando no hay suficiente ancho de banda para aceptar una nueva petición el servidor podrá reducir o modificar, antes de aceptar la reserva, otras reservas semi-elásticas, para liberar suficiente ancho de banda para la nueva reserva. En cualquier caso, los niveles de QoS de todas peticiones se mantienen garantizados. Después de esto, se realiza el procesado normal del mensaje RESV.

Estas extensiones no cambian la operación normal de RSVP. Por tanto, los estados de reserva en los routers intermedios,

debido a sesiones RSVP se refrescan de igual forma que en RSVP.

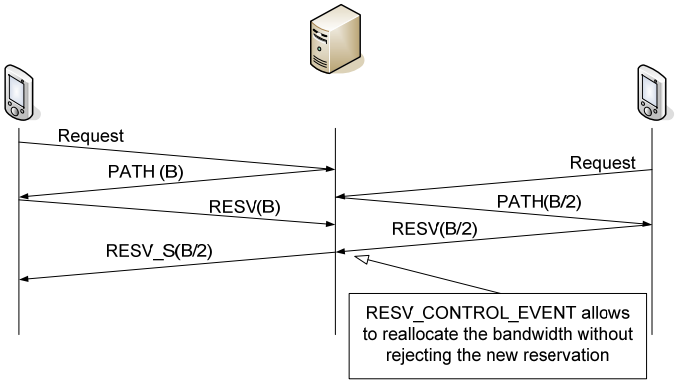


Fig. 1: Extension of RSVP with new messages (RESV_S message).

B. Implementación RSVP

La implementación de las extensiones RSVP en este artículo se realizan usando la versión 4.2a4 de la distribución de RSVP [30] del *Information Science Institute* (ISI). Se basa en este demonio RSVP para permitir la funcionalidad de estos nuevos mensajes y para asegurar la compatibilidad con versiones anteriores del ISI. Sin embargo, añade nuevas funcionalidades para permitir al servidor el modificar las reservas directamente. La distribución del ISI está diseñada para funcionar bajo Linux, FreeBSD, SunOS, Solaris y IRIX y está aceptada como una de las implementaciones de referencia. Además, el interfaz ente el demonio RSVP y las aplicaciones locales se realiza mediante la RAPI (*RSVP Application Programming Interface*) [31]. Brevemente, las funciones que tienen lugar en el procesado de los mensajes son:

1. Procesado de mensajes RESV_S

El mensaje RESV_S se procesa cuando el demonio RSVP llama a la función `accept_resv_s()` y comprueba si existe un estado de PATH previo antes de que se cambie la reserva. Si se confirma, el estado de reserva se establece llamando a la función `flow_reservation_s()`, y los parámetros del RSB (Reservation State Blocks) se comparan y actualizan con los del mensaje RESV_S. Finalmente, la función `Complete_ModFlowspec_s()` se encarga de aplicar la llamada RESV_S_EVENT, avisando a la aplicación local de que la reserva ha cambiado. Si no hay una reserva, se envía un mensaje RESV_S_ERROR y finaliza el proceso.

2. Procesado del mensaje RESV_S_TEAR

El mensaje RESV_S_TEAR se procesa cuando el demonio llama a la función `accept_resv_s_tear()` para confirmar que la sesión del paquete recibido está presente y para poder continuar con el proceso. En caso contrario finaliza.

La función `kill_RSB_s()` se encarga de eliminar la reserva que corresponde al RSB. Finalmente, llama a la función `LL_DelFlow()` que actualiza la función `Complete_DelFlow_s()` y realiza una llamada RESV_S_EVENT cuando se alcanza el receptor.

3. Procesado de mensajes RESV_S_ERR

El procesado de este mensaje es similar al del mensaje RESV_ERR, pero en este caso, es el emisor (servidor) en vez de receptor el que recibe la llamada desde el demonio.

4. Procesado de mensajes RESV

Extendemos el procesado de los mensajes RESV para permitir a las aplicaciones finales la gestión de su ancho de banda. Cuando no hay suficiente ancho de banda para usar el modo ReR en el enlace de acceso del servidor, esta reserva se rechaza y el emisor no puede realizar otras acciones para evitarlo. Mediante esta nueva extensión de RSVP el emisor detecta la llegada de nuevas peticiones de reserva antes de que se establezca en el enlace de acceso del emisor usando la llamada RAPI_RESV_CONTROL_EVENT. Si el emisor detecta que no hay recursos adecuados para acceder esta nueva petición, puede ajustar las otras reservas cambiando sus tasas o cancelando algunas de ellas si esto es aceptable. En cualquier caso, el emisor intenta evitar el mayor número de rechazos posible. La Fig. 2 describe el procedimiento que se sigue en el servidor cuando llega un nuevo mensaje RESV.

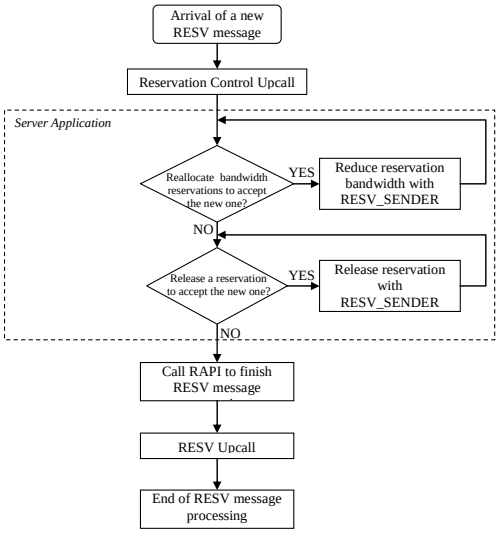


Fig. 2: RSVP extension. The arrival of new RESV message to the server

III. Caso de Aplicación: Flujos Semi-Elásticos

Esta sección presenta un caso de aplicación en el que se usan las extensiones RSVP. Hay otras muchas aplicaciones como las multimedia o las que necesitan transmisión en tiempo real que necesitan algún mecanismo para garantizar un determinado nivel de QoS para funcionar de forma adecuada. Los flujos semi-elásticos pueden usar las reservas de recursos para reducir el tiempo de espera o para garantizar la disponibilidad de datos en el receptor. Debido a la elasticidad de las reservas presentes, estas pueden reducirse o liberarse, si es necesario, sin afectar a la calidad de su transmisión, si el receptor dispone de suficientes datos disponibles.

En nuestra propuesta, el servidor puede reservar recursos para enviar información a una tasa α_{ReR} que será mayor que la que necesita el cliente para leer, para así tener por adelantado más información de la que necesita [29] para así usar el servidor normal BE. Cuando esto ocurre, el cliente deja de recibir más información usando la reserva de recursos. Libera los recursos reservados que pueden usar otros flujos semi-elásticos que acceden al servidor, y continúan recibiendo datos en el modo BE a una tasa α_{BE} . Después, se explicará el algoritmo que gestiona el acceso múltiple de varios clientes al servidor en este caso.

A. Gestión del ancho de banda de acceso

Suponemos que el ancho de banda del servidor B_s b/s, está dividido en tres partes: para el servicio BE (B_{BE} b/s), para los servicios de reserva (B_{ReR} b/s), y para la señalización RSVP (B_{sig}). Además, la transmisión BE se asume que se carga con una tarifa plana, hacienda la transmisión BE más barata que la ReR. Los clientes son homogéneos, tienen la misma tasa de lectura (α_r b/s) y la tasa de reserva α_{Res} es siempre mayor que α_r .

Consideramos tres situaciones diferentes en la gestión del ancho de banda disponible para ReR: iniciar una nueva sesión, pasar a modo ReR y liberar una sesión.

B. Iniciar una sesión

La tasa máxima en el modo ReR depende del número de clientes conectados (*Num_clients)*. Si este número es menor que el número de reservas semi-elásticas simultáneas (*Max_resv)*, α_{Res} decrece con cada nueva conexión ya que se reparte entre todas las conexiones. Cuando *Max_resv* es 1, sólo se permite una conexión en modo ReR. Esto hace que los datos lleguen más rápido al cliente y por tanto, su memoria alcanza el umbral *Max* más rápido y se usará el modo BE durante más tiempo. Cuanto mayor es *Max_resv*, el ancho de banda asignado a las reservas simultáneas decrece, haciendo que los datos recibidos durante BE decrezca. Cuando llegan nuevos clientes, cada uno obtiene la misma parte de B_{ReR}. Si el número de clientes conectados excede *Max_resv*, α_{Res} no varía.

C. Paso al modo ReR

El servidor controla el ancho de banda asignado a cada reserva. Cuando el cliente solicita el paso al modo ReR, el servidor recibe una llamada RAPI_RESV_CONTROL_ EVENT. El servidor sabe que una reserva pendiente espera el acceso y por tanto, debe gestionar el resto de reservas para aceptar la nueva petición ReR. Si el servidor da servicio al número máximo de reservas simultaneas admitidas, envía un mensaje RESV_S_TEAR al cliente más antiguo en el modo ReR para cambiarle a BE y la reserva pendiente se completa. Alternativamente, si el máximo número de reservas simultáneas no se ha alcanzado, el servidor ajusta la nueva tasa de reserva para cada cliente usando el mensaje RESV_S. La Fig. 3 muestra este proceso.

D. Liberar una sesión

Cuando el cliente quiere finalizar su sesión, el servidor actualiza las reservas para cada cliente considerando el nuevo número de clientes conectados. Si los clientes restantes exceden *Max_resv*, la tasa máxima para cada cliente no varía. Por el contrario, la nueva tasa para cada cliente se calcula y se envía un mensaje PATH a cada uno incluyendo este nuevo valor.

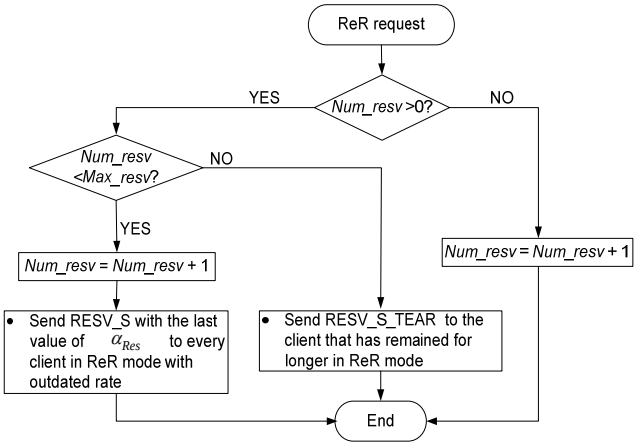


Fig. 3: Management of the server access bandwidth when a client asks to pass to ReR mode.

IV. Resultados de Evaluación

Hemos realizado de forma exhaustiva simulaciones para evaluar el caso de aplicación en Network Simulator 2 (ns-2). Se ha utilizado una implementación de RSVP para ns-2, RSVP/ns [32]. Para este trabajo, se mejora RSVP/ns con los nuevos mensajes y el algoritmo que permite al servidor la gestión de reservas de recursos en el caso de los flujos semi-elásticos. Se utilizan los valores recomendados por defecto para la temporización de mensajes RSVP [1].

La arquitectura multicliente-servidor del la Fig. 4 muestra la topología implementada. Para ser preciso, los clientes leen datos de sus memorias a la misma tasa (600 kb/s), y usan una memoria de tamaño 1 MB. Cada cliente abre sesiones desincronizadas y solicita un archivo de 300 MB. El tráfico de fondo se considera de tasa constante, incluido en la distribución del simulador NS. En media, para el modo BE, los clientes reciben a una tasa de 200 kb/s. El ancho de banda de acceso del servidor es de 10 Mb/s y el ancho de banda asignado para el modo ReR es de 8 Mb/s. Como la tasa de lectura del cliente es de 600 kb/s y el ancho de banda asignado para el modo ReR es 8 Mb/s, el servidor puede aceptar un número máximo de clientes de 13.

Las simulaciones se ejecutan independientemente 10 veces para evaluar los intervalos de confianza del 95%, aunque no se muestran en los resultados debido a que son suficientemente estrechos. Se realizan cerca de 10^5 solicitudes de sesión por cada simulación.

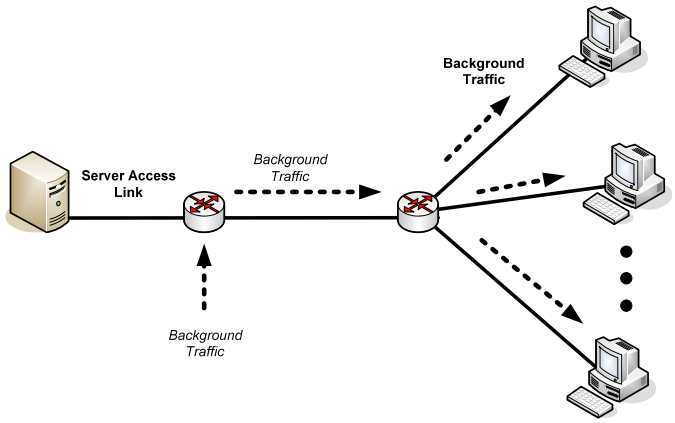


Fig. 4: Simulation and prototype scenario

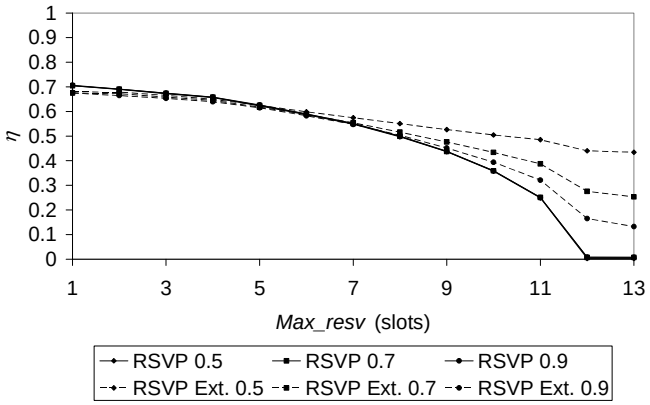


Fig. 5: Efficiency vs. the number of allowed simultaneous ReR connections for different values of offered traffic.

Se utiliza una implementación Tahoe de TCP para transferir los datos, y como los flujos usan TCP, tanto los clientes como el servidor heredan de la clase de ns-2 *TcpApp*, que proporciona el interfaz de aplicación con la capa de transporte TCP. En este caso, necesitamos adaptar el interfaz de *TcpApp* para permitir al servidor el control de la tasa de envío en modo ReR. En el Apéndice C de la referencia [29] se muestra la jerarquía de todas las clases C++ desarrolladas y las tareas necesarias. Los entornos simulados usan scripts OTcl que definen los parámetros adecuados como el número de clientes, los tamaños de los archivos y las tasas de lectura de los clientes.

Hay varias medidas que hemos usado para comparar nuestra versión de RSVP extendida con RSVP nativo: eficiencia, número medio de conexiones activas, probabilidad de bloqueo y señalización extra.

A. Eficiencia

La eficiencia (η) representa el ratio entre los datos enviados usando BE (D_{BE}) y los datos totales ($D_{Total} = D_{ReR} + D_{BE}$). Una eficiencia de 1 significa que todos los datos se han enviado usando BE y una eficiencia de 0 que todos los datos se han enviado usando ReR.

$$\eta = \frac{D_{BE}}{D_{Total}} \tag{1}$$

La Fig. 5 muestra la eficiencia para diferentes valores del número de conexiones ReR simultáneas permitidas (*Max_resv*), cuando la carga ofrecida es de 0.5, 0.7 y 0.9. El comportamiento cuando *Max_resv* es 12 ó 13 muestra el beneficio de redistribuir el ancho de banda entre las conexiones restantes en el servidor. Cuanto menor es el tráfico ofrecido de sesiones, menos conexiones comparten el ancho de banda y consecuentemente, reciben mayores reservas de ancho de banda haciendo posible que aumente la eficiencia. Por el contrario, la eficiencia de RSVP se mantiene igual ya que el ancho de banda reservado por conexión es siempre el mismo y depende de *Max_resv*. La diferencia entre la eficiencia para RSVP extendido y nativo alcanza un 43% para un valor de *Max_resv* de 13.

B. Tasa de sesiones cursadas

Definimos la tasa de sesiones cursadas (*Carried Sessions Ratio*, CSR) como la relación entre las sesiones cursadas por el servidor (S_c) y la cantidad total de sesiones solicitadas por los clientes (S_{Total}).

$$CSR = \frac{S_c}{S_{Total}} \tag{2}$$

Esta tasa puede expresarse en dos partes: la primera puede asociarse al ratio BE cursado (*CbeR*) y la segunda al ratio ReR cursado (*CrerR*). La primera representa el ratio de tiempo que han estado en modo BE las sesiones cursadas y la segunda el de tiempo que han estado en modo ReR.

$$CSR = CbeR + CrerR \tag{3}$$

$$CbeR = CSR \cdot \eta \tag{4}$$

$$CrerR = CSR \cdot (1 - \eta) \tag{5}$$

La Fig. 6 muestra el *CSR* para diferentes valores de *Max_resv*, cuando la carga ofrecida vale 0.5, 0.7 y 0.9. El *CSR* con RSVP extendido se mantiene casi constante. Esto se debe a que las extensiones permiten al servidor aceptar el máximo número de clientes que puede servir. Este número sólo depende de la tasa de lectura del cliente y el ancho de banda de acceso. El tráfico ofrecido de sesiones también afecta al *CSR*. Cuanto mayor es, el *CSR* decrece. Esto se debe a que la cantidad de sesiones bloqueadas también se incrementa. El comportamiento de RSVP mejora cuando aumenta *Max_resv* ya que las sesiones aceptadas se incrementan.

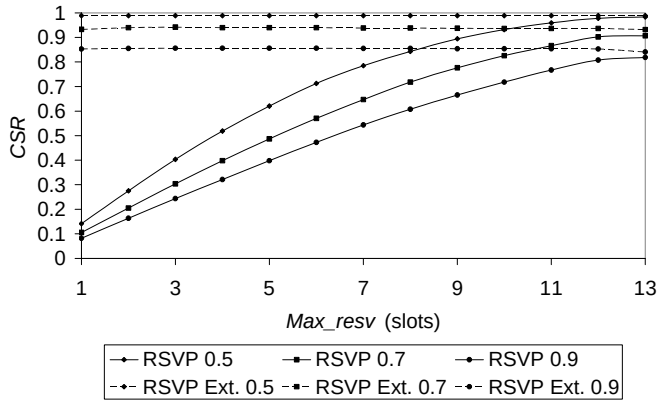


Fig. 6: Carried sessions ratio vs. the number of allowed simultaneous ReR connections for different values of offered traffic.

Las figuras 7 y 8 muestran el *CbeR* y el *CrerR* para diferentes números de conexiones simultáneas aceptadas (*Max_resv*). En cuanto al comportamiento de las extensiones, las conexiones siempre usan más el modo BE que el ReR. También, el modo BE su usa menos a medida que *Max_resv* se incrementa como se explicó anteriormente para la eficiencia. Para RSVP nativo, podemos observar que el modo BE se usa menos que el ReR cuando *Max_resv* toma valores altos ya que con RSVP el servidor no puede redistribuir el ancho de banda.

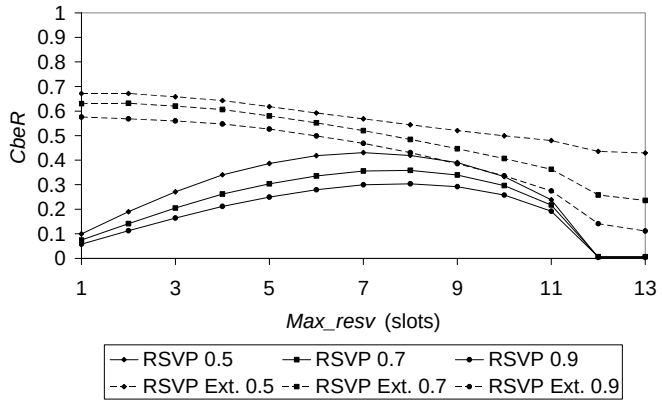


Fig. 7: Carried BE ratio vs. the number of allowed simultaneous ReR connections for different values of offered traffic.

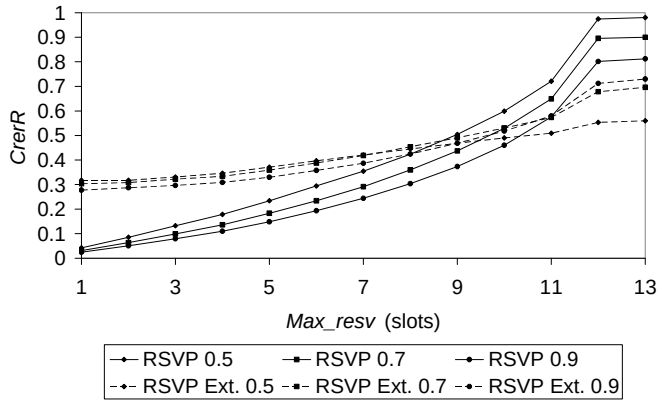


Fig. 8: Carried ReR ratio vs. the number of allowed simultaneous ReR connections for different values of offered traffic.

Comparando el comportamiento de RSVP extendido con el nativo, es destacable que la diferencia máxima del *CSR* se da para *Max_resv* igual a 1. En este caso, las sesiones que cursa un servidor usando RSVP nativo son un 80-90 % menos que con las extensiones (Fig. 6). Por otro lado, si *Max_resv* es igual a 13, el *CSR* para las dos versiones de RSVP dan valores similares, pero mirando a la Fig. 8, podemos observar que el RSVP nativo utiliza ReR durante más tiempo que el extendido.

C. Probabilidad de bloqueo

La probabilidad de bloqueo (P_b) se define como el ratio entre las sesiones cursadas y el total de sesiones recibidas.

$$P_b = 1 - \frac{Carried\_sessions}{Received\_sessions} \quad (6)$$

La Fig. 9 muestra la probabilidad de bloqueo para diferentes tráficos ofrecidos de sesiones, cuando el valor de *Max_resv* es 3, 7 y 11. Como las peticiones de nueva sesión se incrementan, la probabilidad de bloqueo también lo hace. Para RSVP la probabilidad de bloqueo es siempre mayor que para RSVP extendido ya que el *CSR* es menor. Los valores de probabilidad de bloqueo dependen del máximo número de clientes aceptados en el sistema, que se mantiene constante para un valor particular de *Max_resv* con el RSVP extendido. La Fig. 10 representa la probabilidad de bloqueo en función de la eficiencia para diferentes valores de tráfico ofrecido. Los resultados muestran como el RSVP extendió obtiene siempre mejor probabilidad de bloqueo y eficiencia en comparación con el RSVP nativo.

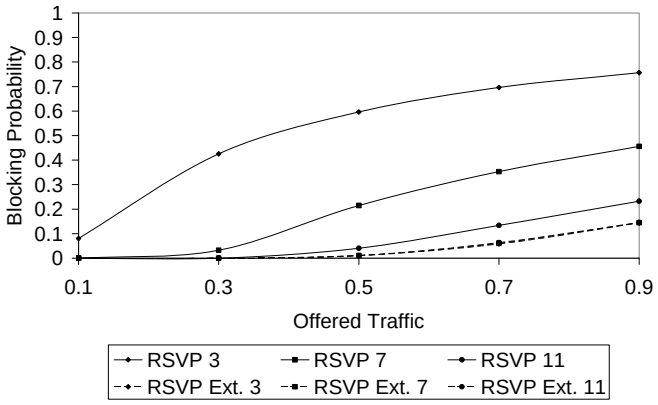


Fig. 9: Blocking probability vs. offered traffic to the server for different numbers of allowed simultaneous ReR connections.

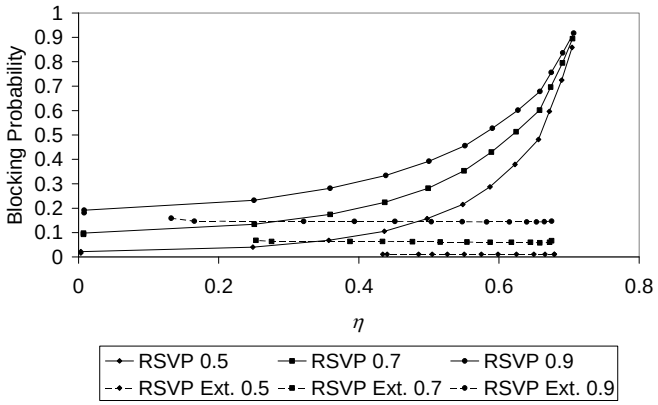


Fig. 10: Blocking probability vs. efficiency for different values of offered traffic.

D. Señalización extra

Medimos las señalización extra necesaria para gestionar el ancho de banda en comparación el RSVP nativo. Son necesarios nuevos paquetes extra cuando se reorganiza el ancho de banda entre sesiones, cuando se acepta una sesión o cuando una sesión finaliza. Además, se necesitan paquetes extra para reorganizar el ancho de banda reservado entre conexiones para no exceder un número de conexiones simultáneas mayor que *Max_resv*.

La Fig. 11 muestra los mensajes de señalización extras por sesión para diferentes valores de *Max_resv*, cuando la carga ofrecida es de 0.5, 0.7 y 0.9. Como se puede ver, para un valor de *Max_resv* igual a 2 la señalización es mínima (aproximadamente 1 mensaje extra por cada 1000 sesiones), lo que clarifica el impacto real de las extensiones. Para *Max_resv* igual a 13, la señalización decrece con el tráfico ofrecido ya que en este caso la señalización extra sólo se utiliza para reorganizar el ancho de banda, cuando el número de sesiones activas varía. La señalización extra se incrementa con *Max_resv* ya que hay un mayor número de conexiones a gestionar.

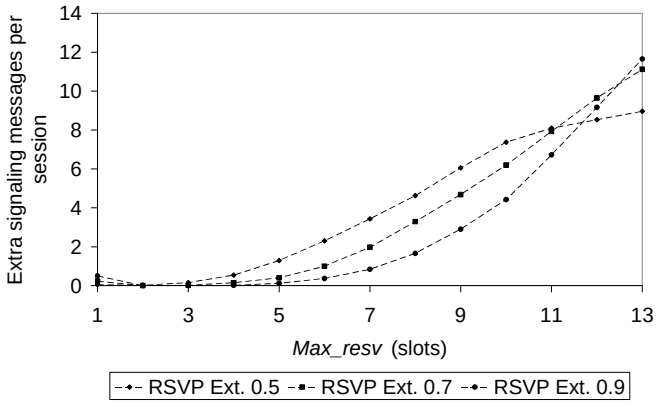


Fig. 11: Extra signaling messages per session vs. the number of allowed simultaneous ReR connections for different values of offered traffic

Asimismo, se ha implementado un prototipo, Fig. 4, para confirmar las simulaciones. Utiliza Linux, lenguaje C++ y threads POSIX. Las aplicaciones utilizan una RAPI (RSVP-API) que incluye las extensiones que interactúan con el demonio extendido RSVP. En este escenario, el servidor se encarga de la gestión del ancho de banda usando las nuevas extensiones de RSVP. Además, puede ajustar o cancelar reservas para ofrecer el servicio al mayor número de flujos posible. Otros elementos son el cliente que se ejecuta en diferentes máquinas usando diferentes terminales y los routers. Se genera tráfico de fondo CBR usando el generador de tráfico MGEN [33]. Los resultados obtenidos son similares a los de simulación lo que confirma la aplicabilidad de la propuesta como un paso intermedio para mejorar la flexibilidad de RSVP.

V. CONCLUSIONES

En este trabajo, se han propuesto nuevas extensiones para el protocolo de señalización RSVP que alertan al servidor de cambios en el estado de las reservas. Esto es útil para ajustar otras reservas, antes de procesar los mensajes RESV que llegan. Se ha analizado el correcto funcionamiento de las extensiones para el caso de flujos semi-elásticos homogéneos que acceden a un servidor. En ese sentido, cuando llegan nuevas reservas al servidor y el ancho de banda disponible está totalmente ocupado, el servidor ajusta las reservas existentes antes de procesar la nueva reserva para evitar su rechazo sin degradar el nivel de QoS necesario por la aplicación, debido a las características de los flujos semi-elásticos. Los valores de las métricas estudiadas confirman una mejora en la eficiencia, el número de sesiones simultaneas cursadas y la probabilidad de bloqueo, con un pequeño incremento en el número de mensajes extra de señalización.

La ventaja de esta propuesta es la habilidad de aceptar más sesiones semi-elásticas que el RSVP nativo, ya que las reservas se tratan de forma más flexible que con RSVP nativo. Esto se confirma con la mejora en el número de sesiones simultáneas cursadas y una menor probabilidad de bloqueo.

AGRADECIMIENTOS

Este trabajo ha sido subvencionado por los proyectos TSI2005-07293-C02-01, TSI2005-06413, y el grupo consolidado de investigación 2005SGR 00563 financiado por la Generalitat de Catalunya.

REFERENCIAS

[1] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, “Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification”. RFC 2205, Sept 1997.

[2] R. Braden, L. Zhang, “Resource ReSerVation Protocol (RSVP) -- Version 1 Message Processing Rules”. RFC 2209, Sept. 1997.

[3] L. Berger, D. Gan, G. Swalow, P. Pan, F. Tommasi, and S. Molendini, “RSVP refresh overhead reduction extensions”, RFC 2961, Apr.2001.

[4] B. Lindell, R. Braden, and L. Zhang, “Resource ReSerVation Protocol (RSVP) -- Version 1 Message Processing Rules [REVISION]”. Internet Draft, February 1999.

[5] M. Karsten, “Experimental Extensions to RSVP – Remote Client and One-Pass Signalling”, in Proc. of the 9th International Workshop on Quality of Service (IWQoS’01), Karlsruhe, Germany 2001, pp. 269-274.

[6] X. Fu and C. Kappler, “Towards RSVP Lite: Light-weight RSVP for Generic Signaling”, Proc. of the 17th International Conference on Advanced Information Networking and Applications (AINA’03), Xi'an, China, March 2003, pp. 619-622.

[7] M. A. Malik, S.S. Kanhere, M. Hassan and B. Benatallah, “ On-Board RSVP: An Extension of RSVP to support Real-Time Services in On-Board IP Networks”, IWDC 2004, LCNS 3326, pp. 264-275.

[8] N-F. Huang and W-En Chien, “RSVP Extensions for Real Time Services in Hierarchical Mobile IPv6”, Mobile Networks and Applications 8, Kluwer Academic, pp. 625-634.

[9] B. Moon and H. Aghvami, “RSVP Extensions for Real-Time Services in Wireless Mobile Networks”, IEEE Communications Magazine Dec.2001, pp. 52-59.

[10] G-S Kuo and Po-Ch. Ko, “Dynamic RSVP protocol”, IEEE Communications Magazine, May 2003, pp.130-135.

[11] Q. Huang and G-S. Kuo “Dynamic RSVP Extension for Wireless Mobile IP Networks”, IEEE 60th Conference on Vehicular technology VTC2004-fall.2004, pp.2683-2687.

[12] S. Yasukawa, J. Nishikido and H. Komura, “Scalable QoS Support mobile Resource Reservation Protocol for Real –time Wireless Internet Traffic”, IEEE GLOBECOM´02, pp. 1475-1479.

[13] A. Mahmoodian and G. Haring, “Mobile RSVP with Dynamic Resource Sharing”, IEEE Wireless Communication and Networking WCNC 2000, pp.896-901.

[14] J. M. Chung, “Analysis of MPLS Traffic Engineering”, Procc. of 43rd IEEE Midwest Symp. On Circuits and Systems 2000, pp.550-553.

[15] Z.H. Xia and Y. Hu, "Extending RSVP for Quality of Security Service", IEEE Internet Computing, M-April 2006, pp. 51-57.
[16] D. Awduche et al., "RSVP-TE: extensions to RSVP for LSP tunnels", RFC 3209, Dec 2001.
[17] C. Curescu and S. Nadjm-Tehrani, "Time-Aware Utility-Based Resource Allocation in Wireless Networks", IEEE Transactions on Parallel and Distributed Systems, vol. 16, no. 7, July 2005.
[18] N. Bhatti and R. Friedrich, "Web Server Support for Tiered Services", IEEE Network, September/October 1999, pp. 64-71.
[19] T. Abdelzaher and N. Bhatti, "Web Server QoS Management by Adaptive Content Delivery", in Proc. of the 7th International Workshop on QoS, May 1999, London, England, pp. 216-225.
[20] S. A. Azad, M. Murshed and L. S. Dooley, "Bandwidth Borrowing Schemes for Instantaneous Video-on-Demand Systems", in Proc. of the 2004 IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, June 2004, pp. 2011-2014.
[21] G. Su and M. Wu, "Efficient Bandwidth Resource Allocation for Low-Delay Multiuser Video Streaming", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 15, No. 9, September 2005.
[22] O. Komolafe and J. Sventek, "An Evaluation of RSVP Control Messsage Delivery Mechanisms", in Proceedings of IEEE Workshop on High Performance Switching and Routing 2004.
[23] O. Komolafe and J. Sventek, "An Evaluation of RSVP Control Messsage Delivery Mechanisms", in Proceedings of INFOCOM 2005.
[24] R. Hancock, G. Karagiannis, J. Loughney and S. Van den Bosch, "Next Steps in Signaling (NSIS): Framework" RFC 4080, June 2005.
[25] J. Manner, G. Karagiannis, A. MacDonald and S. Van den Bosch, "NLSP for Quality-of-Service signaling", Internet draft (draft-ietf-nsis-nltp-07) work in progress, July 2005.
[26] X. Fu, et all., NSIS: A new Extensible IP Signaling Protocol Suite, IEEE Communications Magazine 43 (10), October 2005, pp. 133-141.
[27] Next Steps in Signaling (nsis) Charter, http://www.ietf.org/html.charters/nsis-charter.html.
[28] NSIS Implementation, University of Goettingen, http://user.informatik.uni-goettingen.de/~nsis/news.html
[29] M. Postigo-Boix, J. Garcia-Haro and J. L. Melús-Moreno, "A cost efficient method for streaming stored content in a guaranteed QoS Internet", Computer Networks nº 51, January 2007, pp 309-335.
[30] ISI RSVP Project. http://www.isi.edu/div7/rsvp.
[31] R. Braden and D. Hoffman, "RAPI -- An RSVP Application Programming Interface". Internet Draft, August 1998.
[32] Contributed Code – Nsnam, http://nsnam.isi.edu/nsnam/index.php/Contributed_Code.
[33] Naval Research Laboratory, MGEN - The Multi-Generator Toolset, http://mgen.pf.itd.nrl.navy.mil/

Ampliación de la funcionalidad de la implementación de RTP/RTCP en el simulador NS-2 para soportar la sincronización de grupo multimedia en aplicaciones Cluster-to-Cluster

Fernando Boronat Seguí, Mario Montagud Climent
Departamento de Comunicaciones-Instituto IGIC
Universidad Politécnica de Valencia – Escuela Politécnica Superior de Gandia
Ctra Nazaret-Oliva, S/N, CP 46730, Grao de Gandia
fboronat@dcom.upv.es, mamontor@posgrado.upv.es

***Resumen*- Se presenta una modificación del código de NS-2 relativo a los protocolos RTP/RTCP para poder soportar la sincronización de grupo multimedia en aplicaciones cluster-to-cluster. Existen numerosas aplicaciones de este tipo, con una o varias fuentes de información en un cluster y los receptores en diferentes grupos o clústeres, tales como las de tele-inmersión 3D (3DTI), espacios de trabajo colaborativo por ordenador (CSCWs), presentaciones multimedia distribuidas (DMP), etc. En ellas podemos encontrar requerimientos de transporte muy estrictos debido al uso de múltiples flujos de información interrelacionados. Es necesario un mecanismo de sincronización para conseguir reproducir dichos flujos sincronizadamente, con independencia del número de receptores y de flujos reproducidos en cada cluster. Presentamos la codificación de una solución para la sincronización multimedia de grupo dentro de un mismo clúster y para diferentes clústeres. Adicionalmente, también se presenta una evaluación de la misma en un escenario típico.**

Palabras Clave- Simulador NS-2, RTP/RTCP, Sincronización entre flujos, Sincronización de grupo, sistemas multimedia, Multimedia, aplicaciones cluster-to-cluster, RTP/RTCP.

I. INTRODUCCIÓN

Actualmente, podemos distinguir tres tipos de sincronización multimedia: intra-flujo, inter-flujo y de grupo. La sincronización intra-flujo se ocupa del mantenimiento, durante la reproducción, de la relación temporal dentro de cada flujo, entre las unidades de datos lógicas (LDU) del flujo (por ejemplo, la relación temporal entre las diferentes tramas de una secuencia de vídeo). La sincronización inter-flujo se ocupa de mantener la relación temporal entre los procesos de reproducción de diferentes flujos (por ejemplo, la sincronización labial o *lip-sync*, [1]). La sincronización de grupo, o inter-destinatario (Fig. 1), se ocupa de mantener la reproducción de uno o varios flujos, de forma sincronizada, en varios receptores al mismo tiempo (por ejemplo, los concursos de preguntas en red, o *network quizzes*, donde una misma pregunta multimedia debe presentarse al mismo tiempo a todos los participantes para que gane el primero que conteste acertadamente).

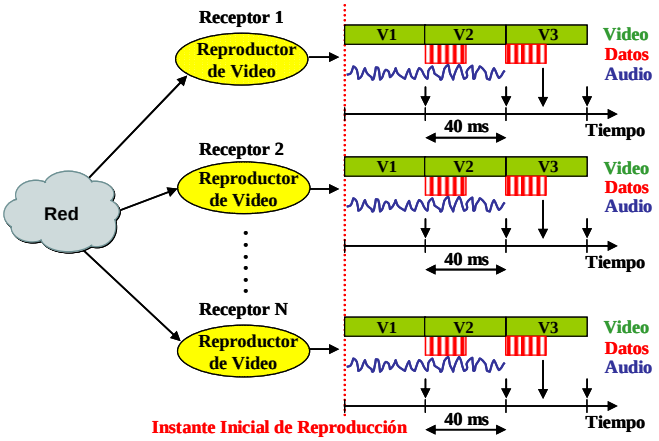


Fig. 1. Sincronización de grupo

El mantenimiento de dichas relaciones temporales normalmente dependerá de los siguientes parámetros que deberán ser considerados individualmente o de una forma integrada ([1]): retardo de red, jitter introducido por la propia red y por los propios sistemas finales, las imperfecciones en los relojes (*skews*, *drifts*…) y las desviaciones en las tasas de generación y de reproducción.

En la actualidad, existen multitud de aplicaciones multimedia *cluster to cluster* (C-to-C) [2]. Por cluster, entendemos una colección de dispositivos de comunicación o sistemas finales que comparten el mismo entorno local. Como ejemplos, podemos citar las siguientes: tele-inmersión 3D (3DTI) [3]; espacios de trabajo colaborativo por ordenador (CSCWs) [4]; presentaciones multimedia distribuidas integrando múltiples flujos (audio, vídeo, texto, datos…) con estrictos requerimientos temporales; entornos ubicuos de computación [5]; etc. En todas ellas existen sofisticados requerimientos de transporte debido al uso de múltiples flujos de información semánticamente relacionados.

Se necesitarán, por tanto, mecanismos de sincronización para subsanar los problemas citados anteriormente, asegurar las relaciones temporales y garantizar la calidad de la aplicación multimedia. Además, el mecanismo de sincronización utilizado deberá ser robusto y adaptarse a los cambios en las condiciones de la red así como a los *drop-outs* de las fuentes, hechos bastante habituales cuando se

utilizan sistemas operativos que no son de tiempo real (Non Real Time Operating Systems, NRTOS).

Los protocolos RTP/RTCP (RFC 3550, [6]) se han convertido en un estándar de facto y son los protocolos de transporte dominantes en la transmisión de datos multimedia.

Los autores han desarrollado una solución para la sincronización inter-flujo y de grupo de flujos multimedia, basada en dichos protocolos y denominada *RTP-based Feedback Global Synchronization Approach (RFGSA)*, descrita en [7] y [8]. En su momento se evaluó en un escenario real pero quedó pendiente la simulación de la solución en otra gran variedad de escenarios. Es por ello que se decidió su implementación en el simulador de redes NS-2 [9].

La implementación de RTP en NS-2 es muy genérica; sólo proporciona las principales funciones de un protocolo de transporte común. En [10] y [11] se dispone de dos modificaciones del código nativo de NS-2 para incluir ciertos aspectos adicionales de RTP/RTCP definidos en la RFC 3550 que no están incluidos en el código nativo del simulador. En [10] las modificaciones están encaminadas a conseguir la implementación de una aplicación para NS-2 capaz de enviar y recibir ficheros de vídeo reales MPEG-2. La implementación de [11] de código RTP/RTCP es más completa que la anterior e incluye, además, un comportamiento *TCP friendly* en cuanto a la adaptación por parte de la fuente de la tasa de transmisión.

En este artículo se presenta otra modificación de dicho código de RTCP para incluir las extensiones a los paquetes RTCP y los nuevos paquetes indicados en [7] y [8], así como nuevas modificaciones indicadas más adelante para poder utilizar estos protocolos en sincronización de grupo en aplicaciones C-to-C unidireccionales. Con estas modificaciones, además de las nuevas funciones de realimentación para medir QoS (jitter, retardo, pérdidas…), las aplicaciones multimedia C-to-C pueden emplear las nuevas extensiones y paquetes RTCP para obtener una sincronización de grupo (inter-destinatario).

Nuestra principal motivación es utilizar el código modificado para las simulaciones en *aplicaciones multimedia C-to-C unidireccionales*, con transmisión multicast unidireccional de flujos multipunto-multipunto, desde una o varias fuentes en un mismo *cluster fuente* hacia varios receptores agrupados en uno o varios *clústeres receptores* según las condiciones de cada uno (Fig. 2). El código NS-2 proporcionará un adecuado marco de trabajo para dichos escenarios de simulación.

El resto del artículo sigue la siguiente estructura. En la siguiente Sección se presentan diversos trabajos relacionados con la sincronización de grupo. En la Sección III resumimos nuestra propuesta de sincronización de grupo. En la Sección IV se presentan unas nociones básicas sobre cómo se implementan los protocolos RTP/RTCP en NS-2 y en la Sección V se detallan algunas de las modificaciones realizadas al código RTP de NS-2 para incluir la propuesta con una modificación para adecuarla a aplicaciones C-to-C. A continuación, en la Sección VI, se presentan los resultados de una simulación de un escenario típico con el nuevo código modificado. Por ultimo, el artículo finaliza con las conclusiones y el trabajo futuro en la Sección VII.

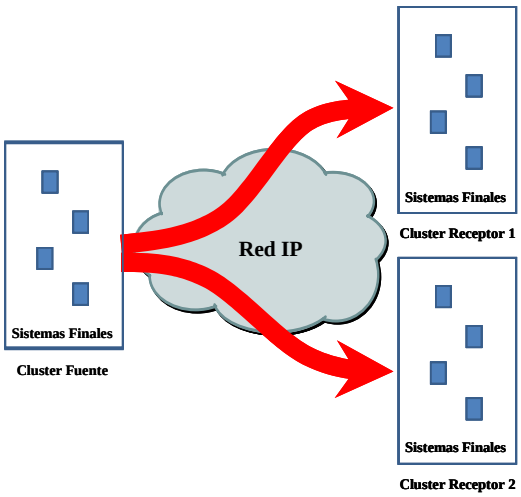


Fig. 2. Modelo de aplicación Cluster-to-Cluster unidireccional

II. TRABAJOS RELACIONADOS

En [12], los autores presentan una recopilación de las técnicas más populares en cuanto a la sincronización de grupo de flujos multimedia, también conocida como sincronización inter-destinatario.

Se pueden distinguir tres técnicas comunes para lograr la sincronización de grupo de flujos multimedia: el esquema maestro/esclavo de receptores (esquema M/E), el esquema de sincronización 'maestro' (esquema ESM) y el esquema de control distribuido (ECD).

En el esquema M/E ([13]), los receptores son clasificados como maestros y esclavos. Cada receptor esclavo no envía información de realimentación sobre los procesos de reproducción. Ellos ajustan su proceso de reproducción de las LDUs al del receptor maestro, que es el único que envía (multicast) dicha información a los demás receptores.

El esquema ESM ([14]) está basado en la existencia de un maestro de sincronización (puede ser la fuente o uno de los receptores), que maneja la información sobre los procesos de reproducción de todos los receptores y les corrige entre ellos mediante la distribución de mensajes de control. Para ello, cada receptor envía información directamente al maestro de sincronización, y éste les envía a todos (multicast) la temporización de reproducción con los ajustes a realizar. Nuestra solución presentada más adelante sigue este esquema, utilizando como protocolo de control a RTCP, y, por tanto, sin definir uno nuevo.

En el esquema ECD ([15]), todos los receptores pueden intercambiar (vía multicast) los paquetes de control o utilizar marcas de tiempo (*timestamps*) en los paquetes de datos multimedia para calcular los retardos de reproducción o *playout delays*, para obtener la sincronización de grupo. Cada receptor decide la referencia en cuanto a la temporización de la reproducción, de entre la suya y la de los otros receptores. En [15] se propone un esquema de control distribuido, que mantiene, de forma adaptativa, las relaciones temporales y causales de acuerdo con la carga de la red bajo un control distribuido. En [16], se presenta el mecanismo de sincronización tipo *bucket* para la sincronización de grupo. En [17], se propone el uso de los algoritmos *local-lag* y *timewarp* para evitar inconsistencias entre usuarios en aplicaciones continuas replicadas (como, por ejemplo, las de juegos en red).

También podemos clasificar los anteriores esquemas de control de la sincronización en centralizados y distribuidos. Los esquemas centralizados tienen sus ventajas y desventajas. Por ejemplo, dichos esquemas pueden preservar la causalidad más fácilmente que los distribuidos. Además, en los esquemas centralizados existe siempre una menor posibilidad de que aparezcan inconsistencias entre los estados de los miembros de la sesión multimedia. Por el contrario, los esquemas centralizados tienen mayores retardos de red y menor fiabilidad y escalabilidad. Por tanto, también podrá ser deseable utilizar el control distribuido para el mantenimiento de la causalidad y de la sincronización multimedia. En [15], se muestran las ventajas y desventajas de los esquemas ESM y ECD en términos de, entre otros, fiabilidad, velocidad y sobrecarga de control. En [18] se mejoraron ambos esquemas para sincronización de grupo, teniendo en cuenta la importancia de los objetos multimedia, para su aplicación en entornos virtuales interconectados en red. En ese trabajo se introduce el concepto de *importancia global* (importancia juzgada desde el punto de vista de todos los usuarios-receptores- en el entorno virtual) junto con el de *importancia local* (importancia juzgada desde el punto de vista de cada usuario, y utilizado para cambiar la precisión de la sincronización intra-flujo e inter-flujo).

En [19] y [20] se mejoran el esquema ESM para la sincronización de grupo, empleado junto con el algoritmo VTR (*Virtual Rendering Algorithm*, [13]), uno de los algoritmos más populares de sincronización intra-flujo e inter-flujo, para ser usados eficientemente en sistemas basados en P2P y en un juego de tiempo real en red con trabajo colaborativo.

Del mismo modo, en [21], los esquemas ESM y ECD, ambos usados para la sincronización de grupo, también son mejorados tomando en consideración la importancia de los objetos multimedia.

En [22], se evalúan los tres esquemas explicados, todos ellos basados en el algoritmo VTR, en una red Multicast Mobile Ad Hoc (*MMAHN*) bastante simple.

Cuando se considera la sincronización de grupo, también se puede considerar un esquema maestro/esclavo pero con respecto a los receptores ([23] y [13]). La temporización de la reproducción del receptor considerado como maestro se toma como la referencia para actualizar la temporización de los otros receptores (esclavos). En algunos algoritmos, el papel de receptor maestro puede cambiarse entre receptores ([23]). En la solución planteada en este artículo, también hacemos uso de este esquema M/S respecto de los receptores.

III. SOLUCIÓN DE SINCRONIZACIÓN DE GRUPO DE FLUJOS MULTIMEDIA

A continuación, vamos a resumir nuestra propuesta anterior para la sincronización de grupo (*RFGSA)*, así como una pequeña modificación de la misma para poder ser utilizada en aplicaciones C-to-C, que dispongan de varios clústeres de receptores (Fig. 2). Se puede obtener más información de la solución preliminar en [7] y [8]. En la Fig. 3 se presenta el proceso de sincronización de grupo propuesto.

Para resolver el problema de la sincronización en los receptores, dividimos el proceso en dos fases: primeramente conseguir que todos los receptores inicien la reproducción de uno de los flujos, considerado como *flujo maestro*, en el mismo instante (*Instante Inicial de Reproducción*); y, a partir de dicho instante, procurar que continúen la reproducción de dicho flujo de forma sincronizada (a este proceso lo denominamos *sincronización distribuida de grupo entre receptores*).

Para ello, nos basamos en dos *esquemas maestro/esclavo*. En la Fig. 3, se puede apreciar la existencia de una transmisión, que puede ser *multicast* o *unicast*, de flujos multimedia mediante RTP desde una o varias fuentes transmisoras a uno o varios receptores (paso 1 en la figura). Uno de los flujos multimedia es tomado como flujo *maestro* (líneas y flechas de mayor grosor) y, además, de entre todos los receptores se selecciona uno de ellos como *receptor maestro* (sombreado en la figura), cuyo estado de reproducción del flujo *maestro* será tomado como referencia para determinar el estado de reproducción de cada uno de los demás receptores (*esclavos*). Este *receptor maestro* podrá ser elegido de varias maneras, según determinados criterios (tal y como se describe en [7] y [8]).

La fuente transmisora del flujo maestro se convertirá en la *Fuente Sincronizadora* (FS) y será la que controlará que la reproducción de los receptores se haga de la forma más sincronizada posible, debiendo procesar y analizar la información de realimentación que éstos le envíen de forma, más o menos, periódica.

Para nuestra propuesta de sincronización de grupo utilizando RTP/RTCP, se propone la modificación de los paquetes de realimentación en RTCP, denominados *RTCP Receiver Reports* (RR, [6]), para incluir la información necesaria para dicho propósito. A este paquete lo llamamos paquete RR EXT (extendido). Los receptores del flujo maestro enviarán estos paquetes de forma más o menos periódica (según lo indicado en [6]) a la FS (paso 2 en la figura 3).

Con la información de realimentación y una estimación de los retardos de la red (que pueden ser obtenidos a partir de la información de tiempo de los propios paquetes RR provenientes de los receptores), la FS puede obtener el estado de reproducción del flujo maestro en todos los receptores activos. Entonces, elegirá a uno de ellos como la referencia o *receptor maestro* y, cuando detecte que la asincronía de algún proceso reproductor de uno de los receptores supere un determinado umbral con respecto al proceso reproductor del receptor maestro (bien porque esté adelantado o retrasado), enviará de forma multicast un mensaje de acción para que los receptores corrijan su desviación respecto del receptor de referencia (paso 3 en la figura 3). Tal y como se puede apreciar, los requerimientos de memoria y de sobrecarga computacional debido al proceso de sincronización de grupo son mínimos.

Como mensajes de acción, la FS hace uso de los paquetes RTCP APP (*Application-defined RTCP packet*), definidos en [6], pero con una extensión dependiente de nuestra aplicación, por lo que los denominamos paquetes APP ACT. La FS los enviará para indicar las correspondientes correcciones a los receptores en sus procesos de reproducción. Es por ello que la solución es *basada en la fuente* (*source-based*). La FS tomará la información que le llegue de los paquetes RR modificados, la procesará y les enviará a los receptores los paquetes de acción pertinentes.

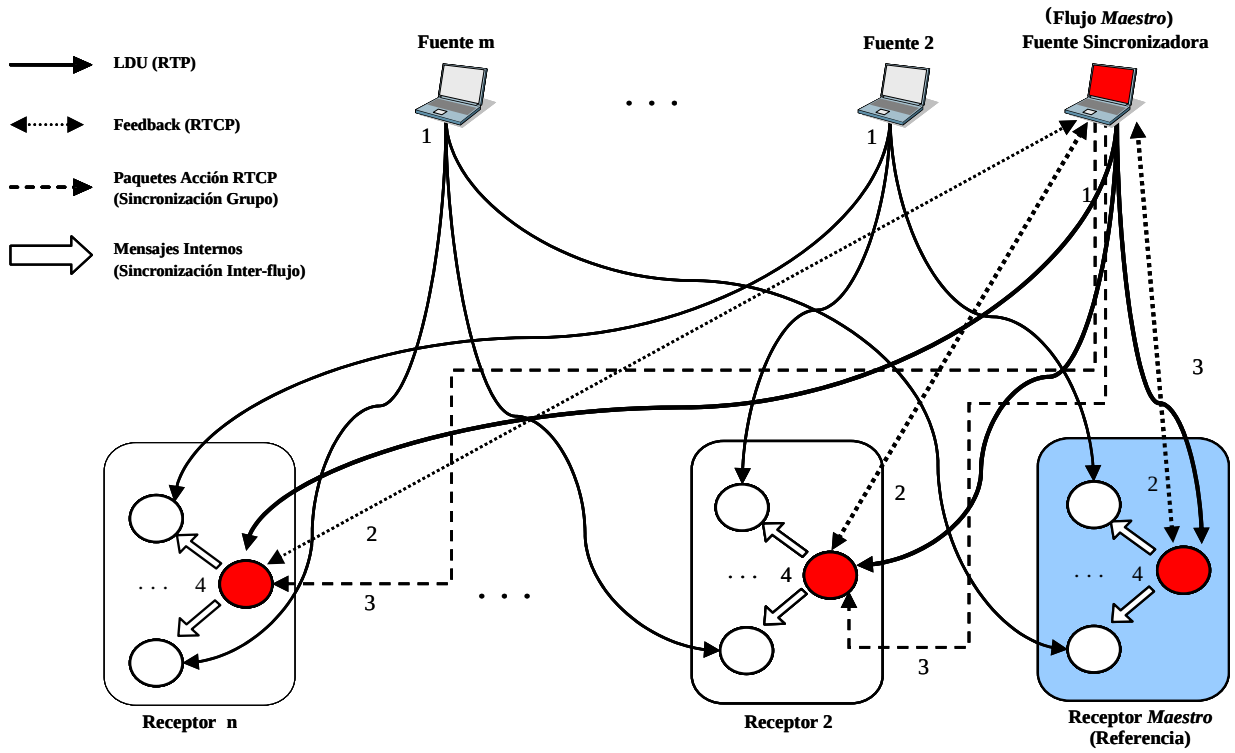


Fig. 3. Sincronización de grupo

Una vez los receptores reciben el paquete APP ACT, y de acuerdo con su contenido, los receptores que estuvieran atrasados o adelantados respecto al proceso reproductor del flujo maestro, adelantarán su reproducción (efecto 'salto') o la retardarán (efecto 'pausa'), respectivamente.

Un paquete con el mismo formato que el APP ACT servirá para indicar el instante de inicio común de la reproducción a todos los receptores de la primera LDU del flujo maestro, con objetivo de conseguir la sincronización del *Instante Inicial de Reproducción*. En el campo de tiempo NTP se incluirá el instante de tiempo global. El proceso de cálculo de dicho instante puede encontrarse en [7].

Una vez conseguida la sincronización de grupo, es decir, cuando ya todos los receptores estén reproduciendo el flujo maestro de forma sincronizada, también será necesario un mecanismo adicional para conseguir la sincronización inter-flujo local (paso 4 en la figura 3), es decir, que, localmente en cada receptor, los flujos que se reproduzcan en el mismo también lo hagan de forma sincronizada entre ellos (la inclusión del modelado del proceso de sincronización inter-flujo en el simulador se deja para un trabajo futuro).

Por tanto, nuestra propuesta de sincronización de grupo está basada en la fuente (*source-based*) y sigue un esquema ESM (explicado en la Sección II). El maestro es la FS que recibe y gestiona la información de los procesos de reproducción de los receptores (incluida en sus paquetes RR EXT) y corrige la temporización de sus procesos de reproducción mediante el envío multicast de los paquetes de control APP ACT.

En la Sección V se presenta la implementación de esta solución en NS-2, con la explicación de las modificaciones o extensiones realizadas a cada paquete, además de una modificación para que sea válida en entornos C-to-C.

IV. RTP/RTCP EN NS-2

Como ya se ha indicado, la implementación de RTP en NS-2 es muy genérica y sólo proporciona las principales funciones de un protocolo de transporte común, por lo que ha sido necesario realizar modificaciones y hacer nuevas aportaciones al código original para incluir nuestra propuesta.

Los protocolos específicos son implementados en NS-2 como Agentes (*Agent*) que son predefinidos en las *clases* particulares. En nuestro caso, los protocolos RTP y RTCP son implementados como las clases *RTPAgent* y *RTCPAgent*, ambas implementadas en C++ en los ficheros *rtp.cc* y *rtcp.cc*, respectivamente. Estos agentes se ocupan de la generación, el envío y la recepción de los paquetes.

Por otro lado, la clase *RTPSession*, definida en el fichero *session-rtp.cc*, es la encargada de gestionar todos lo procedimientos de mantenimiento de toda la sesión RTP, como por ejemplo, el cálculo del intervalo de envío de paquetes RTCP, su confección, el mantenimiento de las tablas de datos de los participantes, etc.

Cuando se crea una sesión, se crean 4 objetos: el agente RTP, el agente RTCP, un agente fuente (*RTPSource*) y un *RTP Timer*. Cuando se une la sesión a un grupo multicast, los agentes RTP y RTCP se unen a los grupos multicast separados. Esta es la solución adecuada cuando se utiliza el mismo grupo multicast pero con diferente número de puerto.

El procedimiento *start* inicializa el agente RTCP, mientras que el procedimiento *transmit* lanza el agente RTP.

El lenguaje utilizado para crear *scripts* de configuración de los escenarios de simulación en NS-2 es el TCL (*Tool Command Language*). A continuación se muestra parte del código del script TCL de una simulación (las líneas iniciadas con '#' son comentarios del autor):

```
#Nodo 0, defino la fuente multicast
set s0 [new Session/RTP]
#Nodo 1, defino un receptor
set s1 [new Session/RTP]
set s2 [new Session/RTP]
#Creo un grupo multicast
set mrthandle [$ns mrtproto $mproto {}]
set group [Node allocaddr]
#Se unen todos al grupo multicast
$ns at 0.1 "$s0 join-group $group"
$ns at 0.1 "$s0 start"
$ns at 0.1 "$s1 join-group $group"
$ns at 0.1 "$s1 start"
$ns at 0.1 "$s2 join-group $group"
$ns at 0.1 "$s2 start"
#La fuente RTP inicia la transmisión de paquetes RTP
$ns at 0.5 "$s0 transmit"
```

En la siguiente Sección se presenta la modificación realizada en el código de NS-2.

V. IMPLEMENTACIÓN DE LA SOLUCIÓN DE SINCRONIZACIÓN EN NS-2 Y MODIFICACIÓN PARA APLICACIONES C-TO-C

A continuación se explica cómo se han implementado las modificaciones propuestas en las librerías del simulador NS-2 (versión 2.33), de acuerdo con nuestra propuesta en [7] y [8] y la modificación para su validez en aplicaciones C-to-C. Para ello, se ha partido de los fuentes del código RTP/RTCP de NS-2 disponible en [11] ya que es el código que hemos encontrado que añade más funcionalidades de las indicadas en la RFC 3550, pero eliminando toda la funcionalidad del comportamiento *TCP friendly* que no nos interesan.

En la extensión propuesta del formato del paquete RR EXT se debe incluir la siguiente información útil para la sincronización de grupo: el número de secuencia de la LDU que el receptor está reproduciendo en el momento del envío del RR EXT y el tiempo NTP del instante (tiempo global) de dicho envío. Para las aplicaciones C-to-C, además, será necesario añadir un campo con un identificativo del clúster (*cluster_id*, de 13 bits) al cual pertenece el receptor, para que la FS sepa a qué cluster pertenece.

A continuación se presentan los campos añadidos a la estructura del paquete en el código de NS-2 (concretamente, en el fichero *rtp.h*).

```
struct receiver_report
{
    …
    ////////////////////////////////////////////////////////////////////////////////
    /*Información para la sincronización de grupo*/
    // NTP timestamp del instante del envío
    //32-bit most significant word
    double ntp_msw_RR;
    //32-bit least significant word
    double  ntp_lsw_RR;
    //Num secuencia de la LDU siendo reproducida
    int seq_LDU_RR;
    //identificador de recepción del RR extendido
    unsigned short  rr_rc:1;
    //Identificativo del clúster del receptor
    unsigned int  cluster_id:13;
    // Los dos últimos bits no se utilizan. Uso futuro
    unsigned short  NotUsed:2;
////////////////////////////////////////////////////////////////////////////////
 }
```

Como ya se ha indicado anteriormente, los receptores del flujo maestro envían estos paquetes de forma más o menos periódica a la FS. En el código de NS-2, cada receptor creará un informe RR EXT de la siguiente manera (en la función de construcción de informes RTCP, en el fichero *session-rtcp.cc*):

```
//añado un receiver report
receiver_report* rr;
rr = new receiver_report;
//relleno el report
rr->cum_pkts_lost()=sp->cum_pkts_lost();
rr->LSR() = sp->LSR();
rr->DLSR()= now - sp->SRT();
rr->jitter() = sp->jitter();
…
////////////////////////////////////////////////
// NTP timestamp del instante del envío
//32-bit most significant word
rr->ntp_msw_RR = NTP_MSW_calculation(NOW);
//32-bit least significant word
rr->ntp_lsw_RR = NTP_MSW_calculation(NOW);
//Número de secuencia de la LDU siendo reproducida
rr->seq_LDU_RR=SeqNum_playing_LDU (NOW);
//identificador de recepción del RR extendido
rr->rr_rc=BitR_Calculation();
//Identificativo del clúster del receptor
rr->cluster_id=ObtainCluster[localsrc_->srcid()];
//los dos últimos bits no se usan. Los ponemos a 0.
rr->NotUsed=0;
////////////////////////////////////////////////
//añado el RR al paquete RTCP packet
rh_->rr_ = rr;
```

El cálculo de los campos especificados en la RFC 3550 ([6]), se realiza de acuerdo con su apéndice A.8.

Además, para poder almacenar la información recibida de cada receptor, al objeto receptor de NS-2 (*RTPReceiver*) le hemos añadido los campos necesarios para que la fuente pueda tener constancia de los valores recibidos (tiempo NTP -64 bits- y número de secuencia de la última LDU reproducida -16 bits- en dicho instante) en el último RR EXT recibido de cada receptor así como el identificativo del cluster (13 bits) al que pertenece dicho receptor (apenas 125 bits en total, de memoria adicional necesaria por cada receptor), tal y como se indica a continuación:

```
class RTPReceiver : public TclObject {
public:
    …
    // indica si se ha recibido RR Ext del receptor
    int RR_received;
    //Información para la sincronización de grupo
    //NTP timestamp del instante del envío
    //32-bit most significant word
    double ntp_msw_RR;
    //32-bit least significant word
    double  ntp_lsw_RR;
    //Num. secuencia de la LDU siendo reproducida
    int seq_LDU_RR;
    //Id. del clúster del receptor
    unsigned int  cluster_id:13;
    …
    }
```

Con esta información de todos los receptores de cada cluster, la FS ya puede aplicar el algoritmo de selección de un receptor maestro en cada cluster y crear el paquete de acción correspondiente a cada cluster.

Por otro lado, los paquetes de acción APP ACT contienen la cabecera típica de un paquete APP más una extensión en la que la FS incluye un número de secuencia de LDU del flujo maestro (16 bits) y el instante de tiempo NTP (64 bits)

en que dicha LDU deberá ser reproducida por todos los receptores de cada cluster. El tamaño de este paquete es de 24 bytes. A continuación se muestra la definición de la estructura del paquete APP (en el fichero *rtp.h*) que, como novedad, para aplicaciones C-to-C deberá incluir un identificativo de cluster (*cluster_id*) para indicar a qué receptores debe afectar el mensaje de acción.

```
struct sender_APP_ACT {
    unsigned int version:2;
    unsigned int p:1;
    unsigned int count:5;
    unsigned int pt:8;
    int length;
    u_int32_t    sender_srcid_;
    char         name[4];    // Valdrá "ACT"
    double       ntp_sec;    // NTP timestamp
    double       ntp_frac;
    int          seq;   //secuencia del últ. paq RTP transmitido
    unsigned short  rr_rc:1;  //id. recepción del RR extendido
    // Tipo de algoritmo de sincr utilizado por FS [7].
    unsigned short  algorithm_type:2;
    unsigned int  cluster_id:13; //id. cluster al que va dirigido
}
```

Como se ha indicado en la Sección III, este paquete también sirve a la FS para indicar a todos los receptores el inicio común de la reproducción. Para ello, en el campo de tiempo NTP se incluirá el instante de tiempo global en el que se debe iniciar la reproducción de la primera LDU del flujo maestro y en el campo *cluster_id* se codificará un '0'.

VI. EVALUACIÓN

Las modificaciones explicadas en NS-2 se han incluido en el código de NS-2 y han sido probadas en el escenario de la Fig. 4. Se trata de un escenario con una única fuente (FS) y dos clústeres de receptores según la topología de la figura. El cluster 1 está formado por 5 receptores y el cluster 2 está formado por 3 receptores. Se ha creado intencionadamente un cuello de botella en el enlace entre los routers 3 y 4, para tener diferentes condiciones en cuanto a retardo y ancho de banda disponible entre la fuente y los dos clústeres de receptores. Durante toda la sesión existe una transmisión de trafico CBR/UDP entre los nodos *Fuente CBR* y *Sumidero CBR*, indicados en la figura, de tal manera que se asegure la existencia de pérdidas en el primer nodo del enlace saturado.

Las características del escenario son las siguientes:

- Todos los enlaces son iguales, de 1,5 Mbps y con un retardo de 10 ms.
- El nodo *Fuente CBR* emite tráfico CBR, paquetes de 1000 bytes, con una tasa de 1 Mbps.
- Los paquetes de datos RTP son de 1000 bytes y se envían con una tasa de 550 Kbps.
- Los paquetes RTCP se envían por los receptores con un intervalo inicial de transmisión de 500 ms.

Para la transmisión multicast de paquetes RTP se ha utilizado el protocolo PIM-DM (*Protocol Independent Multicast-Dense Mode*), llamado con el comando TCL (*Tool Command Language*) "*set mproto DM*".

La duración de la simulación fue de 960 segundos (16 minutos), durante los cuales algunos de los nodos fueron entrando y abandonando la sesión de la manera indicada en la Fig. 5. La fuente inicia la transmisión de flujo RTP en el segundo 30. Los nodos van entrando a la sesión de forma progresiva hasta el minuto 5. En el minuto 7, abandonan la sesión dos nodos del clúster 1 y un nodo del clúster 2. Éstos vuelven a la sesión en los minutos 8 (1 nodo) y 9 (2 nodos). En el minuto 11, abandonan la sesión un nodo del clúster 1 y dos nodos del clúster 2. Éstos vuelven a la sesión en los minutos 12 (2 nodos) y 13 (1 nodo).

Desde el segundo 30, en el que la fuente empieza a transmitir, hasta el segundo 959, en el que termina la sesión, ésta envió paquetes RTP de 1000 Bytes de longitud cada uno, a una tasa de 550 Kbps, con lo que se enviaron 63869 paquetes, según el simulador, hecho que se puede comprobar de la siguiente manera:

$$\frac{\left(\frac{550\times10^{3}\,bits}{segundo}\right)}{\left(\frac{8\times10^{3}\,bits}{paquete}\right)} = 68{,}75\,\frac{paquetes}{segundo}\times 929\,segundos = 63868{,}75\,paquetes$$

Con respecto a la sobrecarga añadida por nuestra solución de sincronización de grupo, podemos apreciar que se envía un número bastante bajo de paquetes de control comparado con los paquetes de datos RTP enviados. La fuente envió 1586 paquetes APP ACT (Fig. 6), de los cuales se comprobó que 769 afectaban al clúster 1 y 817 afectaban al clúster 2. Como era de esperar, se enviaron más al clúster 2 ya que al tener menos receptores la fuente dispone antes de todos los RR EXT de los receptores del clúster y puede enviar el paquete APP ACT. Esta cantidad total de paquetes APP ACT supuso un 2,5% respecto del total de paquetes RTP transmitidos por la fuente (63869). Aunque no se aprecia en la gráfica, se ha comprobado que se envían más paquetes APP ACT cuando hay menos receptores activos ya que se reciben antes los RR EXT de todos ellos.

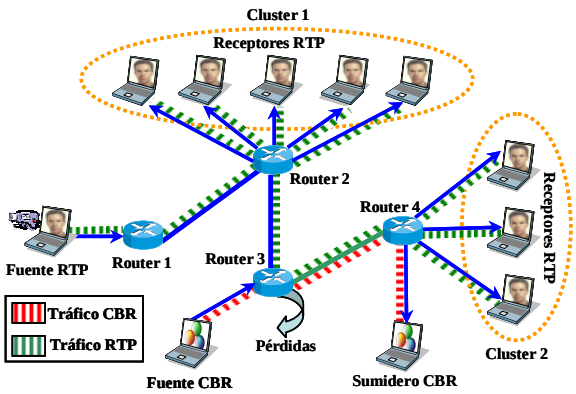


Fig. 4. Escenario de la simulación

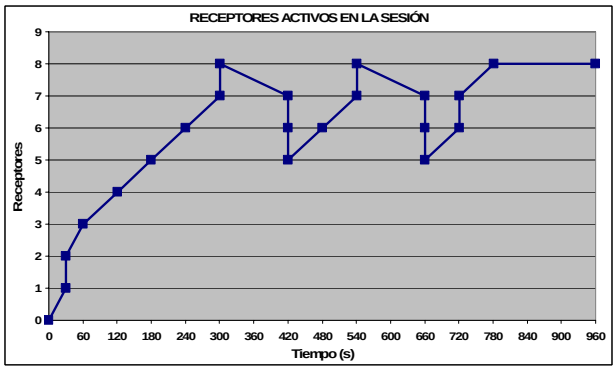


Fig. 5. Receptores activos durante la sesión

Hay que tener en cuenta que esta cantidad es bastante más alta que la que realmente se enviaría en un escenario real

ya que en el caso de la simulación, en estos momentos, se envía un APP ACT siempre que se hayan recibido los RR EXT de todos los receptores de un cluster. En el caso real sólo se enviarían los paquetes de acción cuando la FS detectara una situación de asincronía por encima de un determinado umbral entre los procesos reproductores de los receptores (cálculos que se introducirán en nuestras librerías más adelante). En el escenario real implementado en [8] donde se probó la solución con un único clúster y flujos de audio y vídeo enviados desde una fuente a 10 receptores, la cantidad de paquetes de acción sólo supuso el 1,1% del total de paquetes enviados.

Por otro lado, el número total de paquetes enviados por los receptores fue de 5859 paquetes. En la Fig. 7 aparece el número de paquetes enviado por los receptores. Cabe destacar que este mismo número de paquetes se enviaría tanto si se implementa nuestra solución de sincronización como si no, con la salvedad de que, si se utiliza, cada paquete lleva la extensión adicional anteriormente explicada.

Por tanto, la sobrecarga introducida por nuestra solución sería debida únicamente a los 1586 paquetes de acción APP ACT (de 192 bits cada uno) y la extensión (80 bits) de cada uno de los 5859 paquetes RR EXT enviados durante la sesión. Esta sobrecarga supone un porcentaje muy bajo si lo comparamos con los 63869 paquetes RTP de 1000 bytes cada uno, enviados por la fuente.

En la Fig. 7 se puede apreciar cómo decrece el ritmo de envío de paquetes a medida que algunos receptores abandonan la sesión en los momentos indicados anteriormente y cómo se recupera el ritmo de envío cuando vuelven a entrar a la sesión.

En la Fig. 8 se presenta el número de paquetes RR EXT enviados por cada receptor individualmente. Como era de esperar, se aprecia que los receptores que han estado más tiempo en la sesión son aquellos que más paquetes de realimentación han enviado a la fuente.

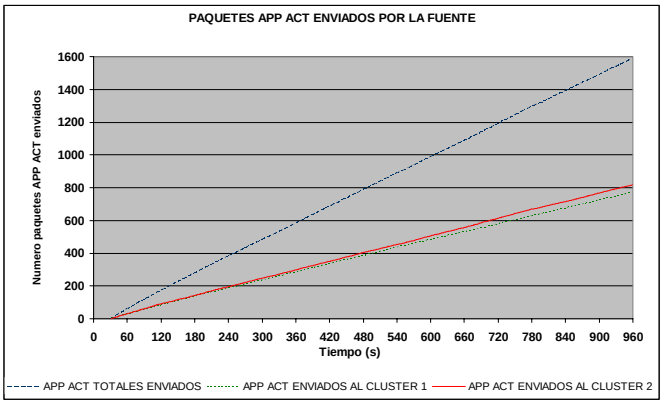


Fig. 6. Paquetes APP de Acción enviados por la fuente

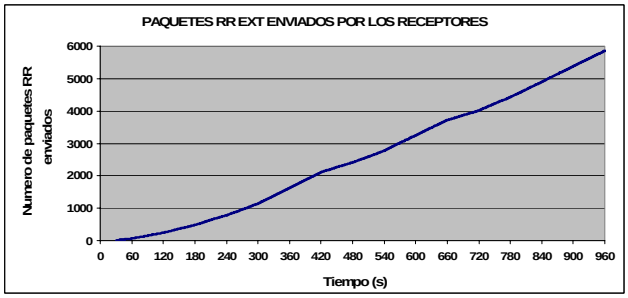


Fig. 7. Paquetes RR enviados por los receptores

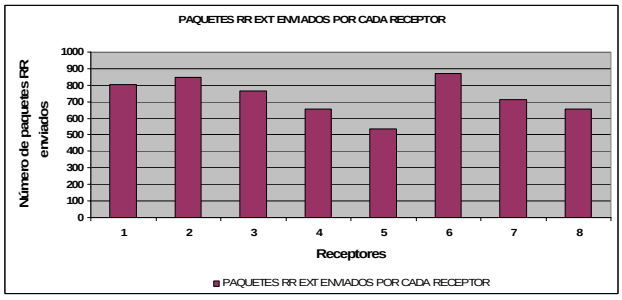


Fig. 8. Paquetes RR Extendidos enviados por cada receptor

VII. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo presentamos una modificación preliminar del código RTP/RTCP de NS-2 para incluir extensiones en los paquetes RR RTCP y poder incluir información de realimentación útil para la sincronización de grupo o inter-destinatario. Las modificaciones se han realizado con el objetivo de incluir una solución propuesta anteriormente por los autores, fruto de un trabajo de investigación previo, modificada para poder utilizarse en aplicaciones Cluster-to-Cluster unidireccionales, de las cuales existen gran variedad hoy en día.

La novedad de la solución de sincronización propuesta por los autores consistía en que no se definía un nuevo protocolo de sincronización, tal y como han hecho otros investigadores. En nuestro caso, se utilizan protocolos estándar utilizados por la mayoría de las aplicaciones multimedia, con lo que la inclusión de la solución en ellas supondría una mínima sobrecarga en cuanto a carga de control de la sincronización.

La implementación de dicha primera parte se ha probado en un escenario típico con resultados satisfactorios en cuanto a la sobrecarga de información de control introducida.

En el momento actual se está en una fase preliminar en la que sólo se ha implementado la parte de modificación del código de RTP en NS-2 relativa a las extensiones de los paquetes y los nuevos paquetes, así como el control de los receptores y su pertenencia a un clúster determinado, dejándose para una etapa futura la definición de los parámetros de las aplicaciones multimedia a simular (por ejemplo, una presentación multimedia a distancia, con flujos de audio, vídeo y datos) y la inclusión de los mecanismos de cálculo de retardos, tiempos de reproducción, y todos aquellos relacionados con la sincronización (asincronías,...).

AGRADECIMIENTOS

El presente trabajo ha sido financiado por la Universidad Politécnica de Valencia bajo el programa de apoyo a la I+D+i en el proyecto 002-585. Los autores también desean agradecer a Víctor Carrascal Frías su ayuda prestada durante los inicios del trabajo en lo referido a la utilización de las librerías RTP/RTCP de NS-2.

REFERENCIAS

[1] Chen, M. 2003. A low-latency lip-synchronized videoconferencing system. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Ft. Lauderdale, Florida, USA, April 05 - 10, 2003). CHI '03. ACM, New York, NY, 465-471.

[2] Manvi S. S. and Venkataram, P. 2006. An agent based synchronization scheme for multimedia applications. Journal of Systems and Software, 79, 5, pp. 701-713, May 2006.

[3] Kum, S-U., Mayer-Patel, K., Fuchs, Z., 2003. Real-time compression for dynamic 3D environments. In Proceedings of the Eleventh ACM International Conference on Multimedia, ACM MM 2003 (Berkeley, CA, USA, November 2-8, 2003), pp. 185 – 194, 2003, ACM Press, New York.

[4] Mortensen, J., Vinayagamoorthy, V., Slater, M., Steed, A., Lok, B., and Whitton, M. C. 2002. Collaboration in tele-immersive environments. In Proceedings of the Workshop on Virtual Environments 2002 (Barcelona, Spain, May 30 - 31, 2002). W. Stürzlinger and S. Müller, Eds. ACM International Conference Proceeding Series, vol. 23. Eurographics Association, Aire-la-Ville, Switzerland, 93-101.

[5] Ott, D. E. and Mayer-Patel, K. 2007. An open architecture for transport-level protocol coordination in distributed multimedia applications. ACM Trans. Multimedia Comput. Commun. Appl. 3, 3 (Aug. 2007), 17.

[6] Schulzrinne, H., Casner, S., Frederick R. and Jacobson V. 2003. RTP: A Transport Protocol for Real-Time Applications. RFC-3550, July 2003.

[7] Boronat, F., 2004. Specification and evaluation of a Multimedia group synchronization algorithm. Doctoral Thesis, Polytechnic University of Valencia (UPV), Spain, ISBN 84-689-0097-4.

[8] Boronat, F., Guerri, J. C. and Lloret, J. 2008. An RTP/RTCP based approach for multimedia group and inter-stream synchronization, Multimedia Tools and Applications Journal, Vol. 40 (2), pp. 285-319, 2008.

[9] Simulador NS-2. http://www.isi.edu/nsnam/ns

[10] Carrascal, V., http://sertel.upc.es/_vcarrascal/ns2/, Grupo de Servicios Telemáticos, Universitat Politécnica de Catalunya, España.

[11] Research Academic Computer Technology Institute, Research Unit 6, University Campus, Building B', Gr-26500 Patras, Grecia, http://Ru6.Cti.Gr/Ru6/Ns_Rtp_Downloads.Php

[12] Boronat F., Lloret, J. and García, M,. Multimedia Group and Inter-Stream Synchronization Techniques: A Comparative Study, Information Systems Journal, Accepted for publication, available on-line, DOI information: 10.1016/j.is.2008.05.001

[13] Ishibashi, Y., Tsuji, A. and Tasaka, S. 1997. A Group Synchronization Mechanism for Stored Media in Multicast Communications. In Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the information Revolution (April 09 - 11, 1997). INFOCOM. IEEE Computer Society, Washington, DC, 692.

[14] Ishibashi, Y. and Tasaka, S. 1997. A group synchronization mechanism for live media in multicast communications. In Conf. Rec. IEEE GLOBECOM' 97, pp. 746–752, November 1997.

[15] Ishibashi, Y. and Tasaka, S. 1999. A distributed control scheme for group synchronization in multicast communications", Proc. of International Symposium Communications, Kaohsiung (Taiwan), pp. 317-323, November 1999.

[16] Diot, C. and Gautier, L. 1999. A Distributed Architecture for Multiplayer Interactive Applications on the Internet. IEEE Network, 13, 4, pp. 6-15, July/August 1999.

[17] M. Mauve, J. Vogel, V. Hilt, and W. Effelsberg. 2004. Local-lag and timewarp: Providing consistency for replicated continuous applications. IEEE Transactions on Multimedia, 6, l, pp. 47-57, February 2004.

[18] Ishibashi, Y. and Tasaka, S. 2002. A distributed control scheme for causality and media synchronization in networked multimedia games. Proc. 11th International Conference on Computer Communications and Networks, Miami (USA), pp. 144- 149, October 2002.

[19] Hashimoto, T. and Ishibashi, Y. 2006. Group synchronization control over haptic media in a networked real-time game with collaborative work. In *Proceedings of 5th ACM SIGCOMM Workshop on Network and System Support For Games* (Singapore, October 30 - 31, 2006). NetGames '06. ACM, New York, NY, 8.

[20] Kurokawa, Y., Ishibashi, Y. and Asano, T. 2007. Group synchronization control in a remote haptic drawing system. Proc. of IEEE International Conference on Multimedia and Expo, Beijing (China), pp. 572-575, July 2007

[21] Ishibashi, Y., Tomaru, K., Tasaka, S., and Inazumi. K. 2003. Group synchronization in networked virtual environments. Proc. Of the 38th IEEE International Conference on Communications 2003, Alaska (USA), 2, pp. 885–890, May 2003.

[22] Nunome, T. and Tasaka, S. 2005. Inter-destination synchronization quality in a multicast mobile ad hoc network. Proc. of IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications, Berlin (Germany), 2, pp. 1366-1370, September 2005.

[23] Akyildiz I. F. and Yen, W. 1996. Multimedia Group Synchronization Protocols for Integrated Services Networks. IEEE Journal Selected Areas in Communications, 14, 1, pp. 162-173, January 1996.

Propuesta de Mecanismos de Negociación para la Optimización en Entornos Altamente no Lineales

Ivan Marsa-Maestre, Miguel A. Lopez-Carmona, Juan R. Velasco y Enrique de la Hoz
Departamento de Automatica, Universidad de Alcala
Edificio Politecnico, Ctra. N-II, Km 36.500
28871 Alcala de Henares (Madrid), Spain
{ivan.marsa,miguelangel.lopez,juanramon.velasco,enrique.delahoz}@uah.es

***Resumen*—La resolución de problemas de optimización distribuida puede abordarse mediante la aplicación del paradigma de las negociaciones automatizadas, principalmente en el caso de escenarios parcialmente observables. Estos problemas de optimización pueden venir descritos por espacios de acuerdo de múltiples atributos interdependientes, y debido a esta interdependencia, por funciones no lineales de preferencia o utilidad. Esta no linealidad hace que los mecanismos de negociación tradicionales no sean aplicables. Incluso mecanismos específicamente diseñados para espacios de utilidad no lineales pueden fallar si el espacio es altamente no lineal. Simulated annealing, por ejemplo, se ha empleado con éxito en negociaciones basadas en subastas, en el caso de espacios de utilidad definidos mediante restricciones. En este artículo, se demuestra empíricamente que la efectividad de este enfoque decrece de forma drástica en escenarios altamente no lineales, y se proponen mecanismos alternativos para el proceso de subasta que tienen en cuenta la estructura del modelo de preferencias. Se propone además un método de búsqueda probabilístico en el mediador que mejora la escalabilidad del proceso de identificación de acuerdos, y un protocolo de negociación expresivo e iterativo para proporcionar realimentación a los agentes si no se encuentran acuerdos en las etapas iniciales. Los experimentos demuestran que los mecanismos propuestos contribuyen a una mejora significativa en la eficiencia de los procesos de optimización para el caso de escenarios altamente no lineales.**

***Index Terms*—sistemas multiagente, negociación multiatributo, espacios de utilidad altamente no lineales**

I. Introducción

La negociación integrativa, como paradigma aplicable a los problemas de optimización multiobjetivo distribuida, tiene como objeto que los agentes que negocian, y que en definitiva desean optimizar sus funciones de utilidad o coste, sean capaces de alcanzar acuerdos o soluciones eficientes desde una perspectiva conjunta [12]. Dentro de este ámbito, existe un interés creciente por el estudio de escenarios de negociación complejos en los que los agentes negocian múltiples atributos dependientes entre sí [9]. Esta dependencia entre atributos genera funciones de utilidad no lineales, contribuyendo a que los mecanismos clásicos utilizados con funciones lineales no sean aplicables. Este artículo propone un conjunto de mecanismos de negociación multilateral mediada, de tipo subasta, aplicables en escenarios de utilidad altamente no lineales. Dichos espacios de utilidad de los agentes se generan para el caso concreto de una estructura de preferencias basada en restricciones ponderadas.

En [8] se propone un mecanismo de subasta, basado en el muestreo aleatorio del espacio de contratos (soluciones potenciales), y en la aplicación de simulated annealing sobre dichas muestras para identificar regiones de alta utilidad. Estas regiones (ofertas) se envían a un mediador que realiza una búsqueda para encontrar solapes entre las regiones de los diferentes agentes. Los experimentos muestran que este enfoque logra una alta efectividad (medida como una alta tasa de optimalidad y una baja tasa de fallo) en el escenario de evaluación descrito por los autores (Sección II). Sin embargo, como se demuestra empíricamente en la Sección VI-B, esta aproximación es ineficiente cuando las circunstancias del escenario se endurecen (esto es, cuando el espacio es altamente no lineal). Bajo esas circunstancias, la tasa de fallos para la estrategia de subasta basada en simulated annealing se incrementa drásticamente, por lo que es necesario un enfoque alternativo para escenarios altamente no lineales, como los que aparecen en interacciones B2B o en sistemas de control automático distribuido.

Además, tal y como se describe en [8], el protocolo de negociación basado en subasta presenta problemas de escalabilidad debido a la búsqueda extensiva de acuerdos que se realiza en el mediador, lo que limita el número máximo de ofertas (pujas) que puede hacer un agente dependiendo del número de agentes que negocian. En este artículo, abordamos estos problemas de la siguiente manera. Proponemos tres mecanismos alternativos frente a simulated annealing para la formación de ofertas basadas en las preferencias de cada agente (Sección III): una búsqueda voraz probabilística, un enfoque basado en programación entera binaria [16], y una búsqueda basada en encontrar conjuntos independientes de peso máximo [1] en el espacio de restricciones de los agentes. Los tres mecanismos evitan el muestreo aleatorio del espacio de contratos, y muestrean directamente el espacio de preferencias (restricciones) del agente, que es considerablemente más pequeño. Además, los mecanismos tienen en cuenta tanto la utilidad de una oferta como su *viabilidad* (una medida de la probabilidad de que la oferta culmine en un acuerdo). Veremos que estos mecanismos de puja mejoran de manera significativa tanto la tasa de optimalidad como la tasa de fallos frente al enfoque de simulated annealing. Se propone además un mecanismo de búsqueda heurística para el mediador que atenúa los problemas de escalabilidad, manteniendo tasas de optimalidad aceptables (Sección IV). Finalmente, se presenta un protocolo expresivo e iterativo para el proceso de negociación, en el que el mediador puede solicitar a los agentes que relajen algunas de sus ofertas para dirigir la negociación a zonas del espacio de contratos donde pueden alcanzarse utilidades conjuntas mayores (Sección V). Este protocolo permite conseguir tasas de fallo aún mejores cuando se combina con los mecanismos de generación de

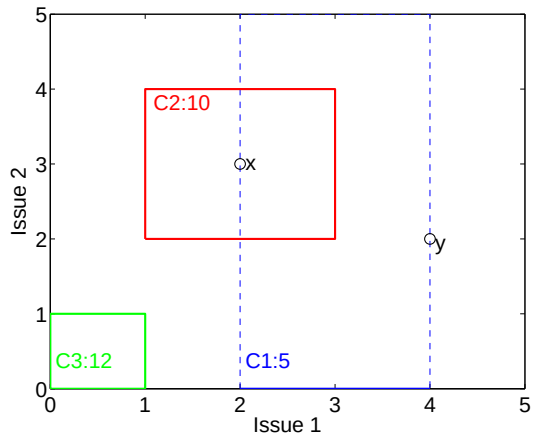


Figura 1. Ejemplo de espacio de utilidad con dos atributos y tres restricciones.

ofertas citados arriba.

Para validar las hipótesis de trabajo y evaluar el efecto de las contribuciones, se ha preparado un escenario de simulación altamente no lineal. Este escenario se describe en la Sección VI, junto con la discusión de los resultados obtenidos. Finalmente, comparamos la propuesta con los trabajos más destacados dentro del estado del arte (Sección VII). La última sección presenta las conclusiones y varias propuestas de líneas de investigación futuras.

II. Búsqueda de Ganancias Conjuntas en Espacios de Utilidad No Lineales

II-A. Espacios de Utilidad no Lineales Basados en Restricciones

Las preferencias no lineales de agentes pueden describirse utilizando diferentes tipos de funciones, como las funciones de utilidad K-aditivas [2], los lenguajes de puja [15], o las restricciones ponderadas [7]. Este artículo se centra en espacios de utilidad no lineales generados mediante restricciones ponderadas. En estos casos, la función de utilidad de cada agente se describe a partir de un conjunto de restricciones. Cada restricción representa una región de una o más dimensiones, y un valor de utilidad asociado. El número de dimensiones del espacio viene dado por el número de atributos que se negocian n, por lo que el número de dimensiones de una restricción debe ser menor o igual que n. La utilidad que una determinada solución potencial (contrato) proporciona a un agente es la suma de los valores de utilidad de todas las restricciones satisfechas por ese contrato. La Figura 1 muestra un ejemplo muy sencillo para dos atributos y tres restricciones: una restricción unaria $C1$ y dos restricciones binarias $C2$ y $C3$. Los valores de utilidad asociados a las restricciones también se muestran en la figura. En este ejemplo, el contrato x daría un valor de utilidad para el agente $u(x) = 15$, ya que satisface tanto $C1$ como $C2$, mientras que el contrato y daría un valor de utilidad $u(y) = 5$, porque sólo satisface $C1$. También puede verse que la restricción unaria $C1$ equivale a una restricción binaria para la que la anchura de la restricción para el atributo 2 es todo el dominio del atributo, por lo que podemos generalizar y decir que todas las restricciones tienen n dimensiones.

De un modo más formal, podemos definir los atributos que se negocian como un conjunto finito de variables $x = \{x_i | i = 1, ..., n\}$, y un contrato (o una posible solución al problema de negociación) como un vector $s = \{x_i^s | i = 1, ..., n\}$ definido por los valores de los atributos. Los atributos toman valores del dominio de los enteros $[0, X]$.

El espacio de utilidad de los agentes se define como un conjunto de restricciones $C = \{c_k | k = 1, ..., l\}$. Cada restricción viene dada por un conjunto de intervalos que definen la región en la que debe estar contenido un contrato para satisfacer la restricción. De este modo, una restricción c se define como $c = \{I_i^c | i = 1, ..., n\}$, donde $I_i^c = [x_i^{min}, x_i^{max}]$ define el valor mínimo y máximo para cada atributo para satisfacer la restricción. Las restricciones así definidas describen regiones hiperrectangulares en el espacio n-dimensional. Cada restricción c_k tiene un valor de utilidad asociado $u(c_k)$.

Un contrato s satisface una restricción c si y sólo si $x_i^s \in I_i^c \forall i$. Por simplicidad de notación, denotaremos esto como $s \in x(c_k)$, queriendo decir que s está en el conjunto de contratos que satisfacen c_k. La utilidad de un agente para un contrato s se define como $u(s) = \sum_{c_k \in C | s \in x(c_k)} u(c_k)$, esto es, la suma de los valores de utilidad de todas las restricciones satisfechas por s. Este tipo de funciones de utilidad produce espacios de utilidad no lineales, con puntos elevados donde se satisfacen muchas restricciones, y regiones bajas donde se satisfacen pocas o ninguna restricción.

II-B. Simulated annealing en Negociaciones No Lineales Mediadas

Ito et al. [8] presentaron un protocolo mediado para operar en espacios de utilidad no lineales generados a partir de restricciones ponderadas. El protocolo consta de los siguientes cuatro pasos:

1. *Muestreo:* Cada agente toma un número fijo de muestras aleatorias del espacio de contratos, empleando una distribución uniforme.
2. *Ajuste:* Cada agente aplica simulated annealing a cada muestra para encontrar un máximo local. El resultado es un conjunto de contratos de alta utilidad.
3. *Generación de ofertas:* Cada agente genera una oferta (puja) por cada contrato ajustado de alta utilidad. Las ofertas se generan como la intersección de todas las restricciones satisfechas por cada contrato. Cada agente envía sus ofertas al mediador, junto con la utilidad asociada a cada una de ellas.
4. *Identificación de Acuerdos:* El mediador emplea búsqueda en anchura con poda para encontrar solapamientos entre las ofertas de los diferentes agentes. Las regiones del espacio de contratos que corresponden con la intersección de al menos una oferta de cada agente se marcan como soluciones potenciales. La solución final es aquella que maximiza la utilidad conjunta, definida como la suma de utilidades para los diferentes agentes.

El protocolo se evalúa en un escenario no lineal para diferente número de agentes y atributos, y alcanza buenos resultados en términos de optimalidad (medida como la relación entre la utilidad de las soluciones encontradas empleando el protocolo, y una solución óptima calculada empleando información completa), y de tasa de fallo (medida como la relación entre el número de negociaciones fallidas y el total de negociaciones).

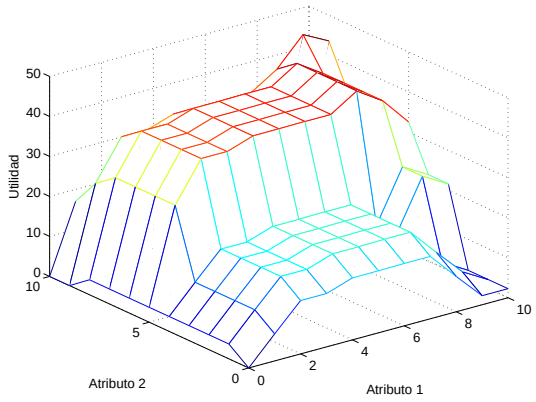


Figura 2. Ejemplo de un espacio de utilidad no lineal generado utilizando restricciones “anchas”.

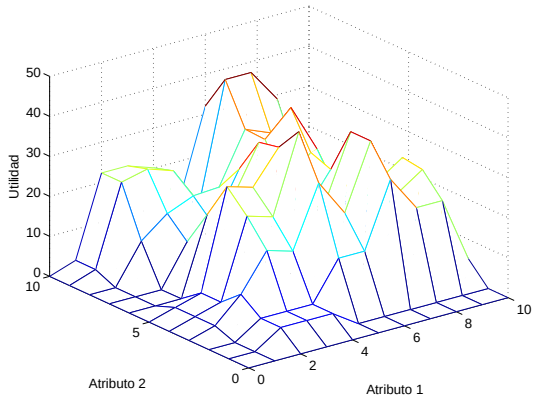


Figura 3. Ejemplo de un espacio de utilidad altamente no lineal generado utilizando restricciones “estrechas”.

El uso de restricciones ponderadas genera un espacio de utilidad “irregular”, con muchos picos y valles. Sin embargo, el grado de “irregularidad” depende en gran medida de la forma en que se genera el conjunto de restricciones, y en particular de la anchura media de las restricciones. En [8], las restricciones se generan escogiendo la anchura de cada restricción para cada atributo de forma aleatoria dentro del intervalo [3,7]. Como el dominio escogido para los atributos es [0,9], esto genera restricciones “anchas”. La Figura 2 muestra un ejemplo del espacio bidimensional que resulta de emplear 50 restricciones binarias generadas de este modo. Por otro lado, la Figura 3 muestra un espacio de utilidad obtenido empleando restricciones “estrechas”, tomando sus valores del intervalo [2,5]. Comparando las dos figuras podemos ver que, aunque ambos espacios de utilidad son no lineales, el espacio generado empleando restricciones estrechas es más complejo, con picos y valles más estrechos. Según aumenta el número de atributos negociados, la diferencia entre usar restricciones anchas o estrechas se vuelve más acusada. Aunque el enfoque propuesto en [8] funciona en escenarios como el ejemplo de la Figura 2, veremos que su desempeño (en términos de las tasas de optimalidad y fallo) empeora drásticamente en escenarios altamente no lineales definidos mediante restricciones estrechas, y por lo tanto es necesario aplicar un enfoque alternativo.

III. Mecanismos de Generación de Ofertas para Espacios de Utilidad Altamente No Lineales

III-A. *Factor de Calidad de una Restricción u Oferta*

Si comparamos los espacios de utilidad de las Figuras 2 y 3, vemos que la principal diferencia entre ellas es la anchura de los picos. Los espacios de utilidad altamente no lineales tendrán picos más estrechos. Como simulated annealing lleva a los agentes a escoger estos picos (o regiones de alta utilidad) como ofertas, el resultado es que al mediador se le enviarán regiones más estrechas. Asumiendo espacios de utilidad generados de forma uniforme, el ancho de las ofertas (o de un modo más general, el volumen de las ofertas en el espacio n-dimensional) influirá directamente sobre la probabilidad de que una oferta se solape con otra de otro agente, y de ese modo influirá sobre la probabilidad de que la oferta resulte en un acuerdo. De forma intuitiva, un agente que no tenga conocimiento acerca de las preferencias de otros agentes deberá equilibrar adecuadamente la utilidad de sus ofertas (para maximizar su propio beneficio) y el volumen de las mismas (para maximizar la probabilidad de éxito en la negociación). Para representar esto formalmente, definimos un *factor de calidad* de una restricción o de una oferta como $Q_c = u_c^{\alpha} \cdot v_c^{\beta}$, donde u_c y v_c son, respectivamente, la utilidad y el volumen de la oferta o restricción c, y α y β son parámetros que modelan la importancia que da el agente a la utilidad final o a la probabilidad de llegar a un acuerdo, respectivamente. Este factor de calidad se utiliza en los mecanismos descritos en las siguientes secciones.

Nuestra hipótesis es que teniendo en cuenta este factor de calidad en los mecanismos de generación de ofertas, con valores adecuados para los parámetros α y β, se obtendrá un mejor equilibrio entre la utilidad y la “anchura” de las ofertas de los agentes, y de este modo las negociaciones darán como resultado tasas de optimalidad más altas y tasas de fallo más bajas.

III-B. *Búsqueda Voraz Probabilística*

El primer mecanismo que proponemos para la generación de ofertas es una ascensión de colina sobre el factor de calidad basada en búsqueda voraz probabilística. En primer lugar, se selecciona una restricción del conjunto. La selección se realiza de tal manera que la probabilidad de que se escoja una restricción es mayor para las restricciones de Q alto. Esta restricción se emplea para generar una oferta inicial b. En cada iteración, se escoge del mismo modo una nueva restricción c de forma aleatoria del conjunto restante, y se calcula su intersección con la oferta b. Si la intersección mejora el factor de calidad, entonces el valor de b se actualiza a esta intersección y el algoritmo itera de nuevo. El algoritmo termina cuando la nueva intersección calculada no incrementa Q o cuando el conjunto restante se vacía. El algoritmo se repite para generar un número fijo de ofertas n_b. El proceso puede verse formalmente en el Algoritmo 1.

III-C. *Programación Entera Binaria y Selección por Torneo*

En [8], el proceso de generación de ofertas (más específicamente, el proceso de muestreo, y ajuste) se trata como un problema de optimización no lineal, en el que el agente busca puntos de alta utilidad en el espacio n-dimensional con

Input:
$C = \{c_k | k = 1, ..., l\}$: conjunto de restricciones del agente
Output: b: nueva puja
$C' = C$;
$b = extract\_random(C')$;
while $C' \neq \emptyset$ **do**
  $c = extract\_random(C')$;
  $b' = b \bigcap c$;
  if $Q_{b'} > Q_b$ **then**
    $b = b'$
end

Algorithm 1: Puja mediante búsqueda voraz probabilística

dominio $[0, X]$. Si, por ejemplo, consideramos un problema de negociación con 10 atributos donde los atributos toman valores del dominio de enteros $[0, 9]$, esto produce un espacio de 10^{10} contratos posibles, lo que hace imposible la evaluación exhaustiva y hace necesario utilizar técnicas heurísticas como simulated annealing. Sin embargo, el espacio de utilidad de los agentes no es arbitrario, sino que ha sido generado empleando un conjunto finito de restricciones ponderadas. Podemos tener esto en cuenta para transformar el problema de optimización en otro diferente contemplándolo desde otra perspectiva.

El proceso de generación de ofertas no es la búsqueda de un contrato, sino la búsqueda de un subconjunto del conjunto de restricciones C que satisfaga dos propiedades:

1. El conjunto maximiza la suma de los valores de utilidad de las restricciones que lo forman.
2. La intersección de todas las restricciones del conjunto no puede ser vacía.

Como cada restricción del conjunto $C = \{c_k | k = 1, ..., l\}$ puede seleccionarse como parte del subconjunto de la oferta, la selección de restricciones puede expresarse como un vector binario $b = \{b_k | k = 1, ..., l; b_k \in [0, 1]\}$, donde $b_k = 1$ si la restricción c_k está incluida en el subconjunto oferta, y $b_k = 0$ en caso contrario. La función de utilidad puede reformularse como $u(s) = \sum_{1 \leq k \leq l} u(c_k) \cdot b_k$, que es una función lineal en un espacio$l$-dimensional de dominio $[0, 1]$. Por supuesto, no todos los vectores b son posibles, ya que la intersección de las restricciones en la oferta no puede ser vacía. Para restricciones hiperrectangulares, esta condición puede asegurarse añadiendo las siguientes inecuaciones al problema:

$$b_i + b_j \leq 1 \; \forall i, j | c_i \bigcap c_j = \emptyset$$

Con esta formulación, se está definiendo un problema clásico de programación entera binaria (*binary integer programming*, BIP) [16], que puede resolverse utilizando, por ejemplo, un algoritmo de búsqueda en árbol con ramificación y poda basado en LP [10]. Sin embargo, esta reformulación del problema no constituye por sí misma una solución viable, ya que tiene algunos inconvenientes serios:

1. Los problemas de programación entera binaria son NP-completos.
2. La cardinalidad del espacio de soluciones es 2^l, que para un número de restricciones elevado puede ser tan intratable como la búsqueda exhaustiva de contratos.
3. El algoritmo de ramificación y poda basado en LP es determinista, por lo que para un determinado conjunto de restricciones obtendríamos siempre la misma oferta.
4. Como para calcular la función de utilidad sólo se emplea la utilidad de las restricciones (y no su factor de utilidad), la oferta generada sería el máximo global del espacio de utilidad, con lo que probablemente presentaría el mismo problema de "anchura" que aparecía con simulated annealing.

Las consideraciones de complejidad computacional pueden resolverse limitando el número máximo de nodos que el algoritmo visita en la búsqueda en árbol, o el número máximo de iteraciones que se realizan en cualquier nodo. Esto, sin embargo, no resuelve el problema de que el algoritmo genere sólo una oferta. Tampoco resuelve el problema de la "estrechez" de la oferta. Para abordar estos problemas, proponemos usar una selección por torneo (*tournament selection*) [14] basada en el factor de calidad Q, es decir, aplicar el enfoque de programación entera binaria a un subconjunto de restricciones $C' = \{c'_k | k = 1, ..., n_c; n_c < l; c'_k \in C\}$. Las restricciones c'_k se escogen de la siguiente manera: se generan de forma aleatoria n_c subconjuntos del conjunto de restricciones C con cardinalidad n_c, y se selecciona aquél que maximice el producto de factores de calidad de sus restricciones, de tal manera que las restricciones de Q alto tienen mayor probabilidad de ser escogidas. De este modo, el algoritmo recibirá como parámetro en cada ejecución un subconjunto de restricciones C' distinto, lo que dará como resultado ofertas diferentes y no deterministas. Además, como es más probable que se seleccionen restricciones de Q elevado, la anchura media de las ofertas será mayor.

III-D. Conjuntos Independientes de Peso Máximo y el Algoritmo Max-product

El espacio de utilidad basado en restricciones de un agente puede verse también como un grafo ponderado no dirigido. Consideremos de nuevo el ejemplo de espacio de utilidad de la Figura 1. Pensemos en cada restricción como un nodo del grafo, con un peso asociado que es el valor de utilidad asociado a la restricción. Ahora conectemos todos los nodos cuyas restricciones correspondientes sean *incompatibles*, es decir, que tengan intersección vacía. El grafo resultante se muestra en la Figura 4.

Encontrar la oferta de mayor utilidad en este tipo de grafo puede verse como encontrar el conjunto de nodos no conectados que maximice la suma de sus pesos. Como sólo los nodos incompatibles están conectados, las restricciones resultantes tendrán intersección no vacía. En el ejemplo, esto se conseguiría tomando el conjunto $\{C1, C2\}$. El problema de encontrar un conjunto de nodos no conectados con peso máximo es un problema bien conocido llamado conjunto independiente de peso máximo (*maximum weight independent set*, MWIS). Aunque los problemas MWIS son también NP-completos, en [1] se utiliza un algoritmo de paso de mensajes para estimar el conjunto MWIS. El algoritmo, que es una reformulación del algoritmo clásico max-product llamada "min-sum", funciona como sigue:

1. Inicialmente ($t = 1$), cada nodo i envía su peso ω_i a

sus vecinos $N(i)$ como mensajes.

$$m^1_{i\to j} = \omega_i \forall j \in N(i)$$

2. En cada iteración t, cada nodo i actualiza el mensaje que envía a cada vecino j restando de su peso ω_i la suma de mensajes recibidos de *todos* los vecinos *excepto* j. Si el resultado es negativo, se envía un cero como mensaje.

$$m^t_{i\to j} = \max(0, \omega_i - \sum_{k\neq j, k\in N(i)} m^{t-1}_{k\to i})$$

3. Una vez recibidos los mensajes, un nodo es incluido en la estimación del conjunto $MWIS$ si y sólo si su peso es mayor que la suma de todos los mensajes recibidos de sus vecinos.

$$MWIS^t = \{i | \omega_i > \sum_{k\in N(i)} m^t_{k\to i}\}$$

4. Los pasos 2 y 3 se repiten hasta que el algoritmo converge o se alcanza el número máximo de iteraciones.

Podemos seguir con facilidad los pasos del algoritmo para el grafo de ejemplo de la Figura 4:

1. $t = 1 \Rightarrow m^1_{1\to 3} = 5$, $m^1_{2\to 3} = 10$, $m^1_{3\to 1} = m^1_{3\to 2} = 12$.
2. $t = 2 \Rightarrow m^2_{1\to 3} = 5$, $m^2_{2\to 3} = 10$, $m^2_{3\to 1} = 2$,$m^2_{3\to 2} = 7$.
3. Teniendo en cuenta los mensajes recibidos,

$$MWIS^2 = \{1, 2\}$$

4. $t = 3 \Rightarrow m^3_{1\to 3} = 5$, $m^3_{2\to 3} = 10$, $m^3_{3\to 1} = 2$,$m^3_{3\to 2} = 7$.
5. Teniendo en cuenta los mensajes recibidos,

$$MWIS^3 = \{1, 2\}$$

6. Como $MWIS$ ha convergido, el algoritmo termina.

Aplicar directamente este algoritmo al proceso de generación de ofertas plantea las mismas consideraciones que el enfoque de programación entera binaria. Cuando el número de nodos en el árbol es alto, el número de iteraciones que necesita el algoritmo para converger puede ser muy elevado. De nuevo, el algoritmo es determinista, por lo que sólo puede generarse una puja para un conjunto de restricciones determinado. Además, este enfoque tampoco considera el volumen de las restricciones. Teniendo esto en cuenta, proponemos combinar este algoritmo con la selección por torneo basada en Q mencionada en la sección anterior, y de este modo aplicar el algoritmo a un subconjunto de restricciones diferente generado aleatoriamente para crear cada oferta.

IV. Un Mecanismo Probabilístico para la Identificación de Acuerdos

La escalabilidad es uno de los principales problemas del protocolo de negociación mediado descrito en [8]. Una vez que los agentes han hecho sus ofertas, el mediador realiza una búsqueda exhaustiva de solapamientos utilizando una búsqueda en anchura con poda. En el peor de los casos, esto significa probar un total de $n_b^{n_a}$ combinaciones de ofertas, donde n_b es el número de ofertas por agente, y n_a es el número de agentes que negocian. En los experimentos, los autores limitan el número de combinaciones a $6,400,000$.

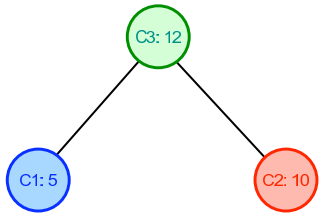


Figura 4. Grafo ponderado no dirigido para el ejemplo de la Figura 1.

Esto significa que, para negociaciones de 4 agentes, el máximo número de ofertas por agente es $\sqrt[n_a]{6400000} = 50$. Este límite se vuelve más estricto al aumentar el número de agentes. Por ejemplo, para 10 agentes, el límite es de 4 ofertas por agente, lo que reduce drásticamente la probabilidad de llegar a un acuerdo. Esto es especialmente cierto para espacios de utilidad altamente no lineales, en los que las ofertas son más estrechas.

Para solucionar esta limitación de escalabilidad, proponemos realizar una búsqueda probabilística en el mediador en lugar de una búsqueda exhaustiva. Esto significa que el mediador probará un cierto número n_{bc} de combinaciones de ofertas escogidas aleatoriamente, donde $n_{bc} < n_b^{n_a}$. De este modo, n_{bc} actúa como un parámetro de rendimiento en el mediador, que limita el coste computacional de la fase de identificación de acuerdos. Por supuesto, restringir la búsqueda de soluciones a un número limitado de combinaciones de ofertas puede hacer que el mediador no pruebe combinaciones que darían lugar a acuerdos provechosos. Teniendo esto en cuenta, la selección aleatoria de combinaciones se sesga para aumentar la probabilidad de encontrar buenos acuerdos. Una vez más, el parámetro empleado para dirigir la selección aleatoria es Q, de manera que las ofertas con Q elevado tendrán mayor probabilidad de ser seleccionadas para las combinaciones de pujas que pruebe el mediador.

V. Un Protocolo de Negociación Expresivo e Iterativo

En espacios de utilidad altamente no lineales, uno de los principales problemas de la negociación básica basada en subasta es que utiliza un protocolo de un solo disparo. Los agentes envían sus ofertas al mediador, el mediador busca soluciones y la negociación termina. Si se ha encontrado una solución, la negociación tiene éxito. Si no, la única posibilidad es repetir el proceso hasta que tenga éxito. En escenarios con regiones de alta utilidad "anchas" esto no es un problema, ya que la probabilidad de que el mediador encuentre una solución es alta. En escenarios altamente no lineales, por contra, como las regiones de alta utilidad son más estrechas, es más probable que una única iteración del protocolo no llegue a una solución. En estos casos, sería deseable que los agentes tuvieran una manera de "aprender" de las interacciones previas para emitir ofertas que tengan mayor probabilidad de éxito. Para que esto sea posible, hacen falta dos mecanismos: uno que permita al mediador proporcionar realimentación a los agentes, y otro que permita a los agentes utilizar esta

realimentación en la generación de ofertas.

Nuestra propuesta es dotar al mediador de capacidad expresiva mediante *requerimientos de relajación*, que expresan qué ofertas debería relajar (o ensanchar) un agente para incrementar la probabilidad de llegar a un acuerdo. Un *requerimiento de relajación* se define como $\rho_{req} = \{b_i | i = 1, ..., p; p \leq n_b; b_i \in B\}$, donde B es el conjunto de ofertas enviadas por el agente y b_i son las ofertas que se le pide que relaje. Estas ofertas se seleccionan calculando el *volumen de acuerdo* δ de cada oferta, que se define como el volumen que debería tener para, asumiendo que su centro permanece inalterado, tener intersección no vacía con al menos una oferta de cada uno de los otros agentes. Una vez calculado el volumen de acuerdo de cada oferta, aquellas cuyo δ esté por debajo de un determinado umbral se incluyen en el requerimiento de relajación.

Una vez que los agentes han recibido los requerimientos de relajación, el proceso de relajación de ofertas comienza. Pueden emplearse diferentes estrategias para relajar las ofertas especificadas por el mediador. Para este trabajo, hemos empleado una estrategia sencilla de concesión mínima. Un agente negociador relaja una oferta eliminando de la misma la restricción que proporciona menor utilidad. Esto puede expresarse formalmente definiendo la oferta relajada como $b' = \{\bigcap\{c_k\} | k = 1...n_c^b, c_k \in b, k \neq j, u_j = \min(u_i | i = 1...n_c^b)\}$. El nuevo conjunto de ofertas está constituida por las ofertas relajadas y por nuevas ofertas generadas hasta completar el número máximo de ofertas n_b.

Resumiendo, el nuevo protocolo expresivo e iterativo consiste en los siguientes pasos:

1. *Puja*: Cada agente a genera un conjunto B^a de n_b ofertas, mediante uno de los mecanismos de la Sección III.
2. *Identificación de acuerdos*: El mediador emplea el método de búsqueda probabilística de la Sección IV para encontrar solapamientos entre las ofertas de los agentes. Si se encuentra una solución, el protocolo finaliza.
3. *Realimentación*: El mediador construye los requerimientos de relajación ρ^a para cada agente y se los envía.
4. *Puja Adaptativa*: Cada agente crea un nuevo conjunto de ofertas B'^a teniendo en cuenta la realimentación recibida.

Los pasos 2 a 4 se repiten hasta que se encuentre una solución o expire un determinado límite (ya sea de tiempo o de número de iteraciones).

VI. Evaluación Experimental

La hipótesis de este trabajo es que los mecanismos propuestos suponen una mejora en términos de optimalidad y tasa de fallos con respecto al trabajo previo descrito en la sección II-B. Para evaluar esto, hemos reproducido los experimentos realizados en [8], comparando los resultados del enfoque previo con los de los mecanismos propuestos.

VI-A. Escenario Experimental

Se han realizado varios experimentos para validar nuestras hipótesis. En cada experimento, ejecutamos 100 negociaciones entre agentes con funciones de utilidad generadas aleatoriamente. Cada negociación se repitió ocho veces para las mismas funciones de utilidad: una para el enfoque basado en simulated annealing, una para cada uno de los mecanismos de oferta propuestos, combinados con el mediador probabilístico, y una para cada uno de los mecanismos de oferta (incluido simulated annealing) empleando el protocolo expresivo.

Para cada conjunto de funciones de utilidad se ha aplicado un optimizador no lineal a la suma de las funciones de utilidad de todos los agentes para encontrar el contrato óptimo y su valor de utilidad conjunta asociado, descartando aquellos contratos con utilidad por debajo de un valor de reserva dado r_v para cualquiera de los agentes. Este contrato óptimo se ha utilizado para valorar la optimalidad de los diferentes enfoques.

Los experimentos fueron ejecutados con los siguientes parámetros:

- Número de agentes n_a = {4,...,10}. Número de atributos n = {4,...,10}. Dominio para los atributos $[0, 9]$.
- l restricciones para cada agente: 5 restricciones unarias, 5 restricciones binarias, 5 restricciones ternarias, etc.
- La utilidad para cada restricción m-aria se toma de una distribución uniforme en el dominio $[0, 100 \times m]$.
- El ancho de cada restricción para cada atributo se toma de una distribución uniforme en el dominio $[2, 5]$.
- Parámetros para simulated annealing: temperatura inicial $T_0 = 30$. Número de iteraciones: 30.
- Número de pujas para cada agente $n_b = 200 \times n$.
- Parámetros para el cálculo de Q: $\alpha = 1, \beta = 1$.
- Número de restricciones tomadas para la selección por torneo: $n_c = \min(20, l/2)$
- Número de combinaciones de ofertas en el mediador: $n_{bc} = 6400000$. Para el mediador de Ito, esto se consigue limitando el número de ofertas que cada agente envía al mediador a $\sqrt[n_a]{6400000}$.
- Número de iteraciones del protocolo expresivo: 1 a 7.
- Valor de reserva del optimizador empleado para calcular la solución óptima: r_v=100.
- Utilidad de una negociación fallida: 0.

Los experimentos se codificaron en Java y se ejecutaron en un procesador Intel Xeon 2x3.2 Ghz Qad-Core con 4Gb de RAM empleando Mac OS X 10.5.4.

VI-B. Resultados Experimentales

La Figura 5 muestra los resultados de los experimentos de una sola iteración. Cada gráfica presenta un diagrama de cajas para los resultados finales de 100 ejecuciones del experimento. El eje horizontal muestra el enfoque evaluado: simulated annealing (*simulated annealing*, SA), búsqueda voraz probabilística *greedy search*, programación entera (BIP) con selección por torneo y conjuntos independientes de peso máximo (MWIS) con selección por torneo. En el eje vertical hemos representado la tasa de optimalidad como un diagrama de cajas y bigotes. Las cajas tienen líneas horizontales para las medianas y los percentiles 25 y 75 de la tasa de optimalidad de las negociaciones (calculada como la relación entre la utilidad conjunta final y la utilidad conjunta óptima), y los bigotes muestran valores adyacentes en los datos. Los elementos atípicos se representan con un signo más (+). Las muescas muestran la variabilidad de la mediana entre muestras. Puede observarse que, aunque el enfoque basado en simulated annealing alcanza a veces tasas de optimalidad elevadas, la

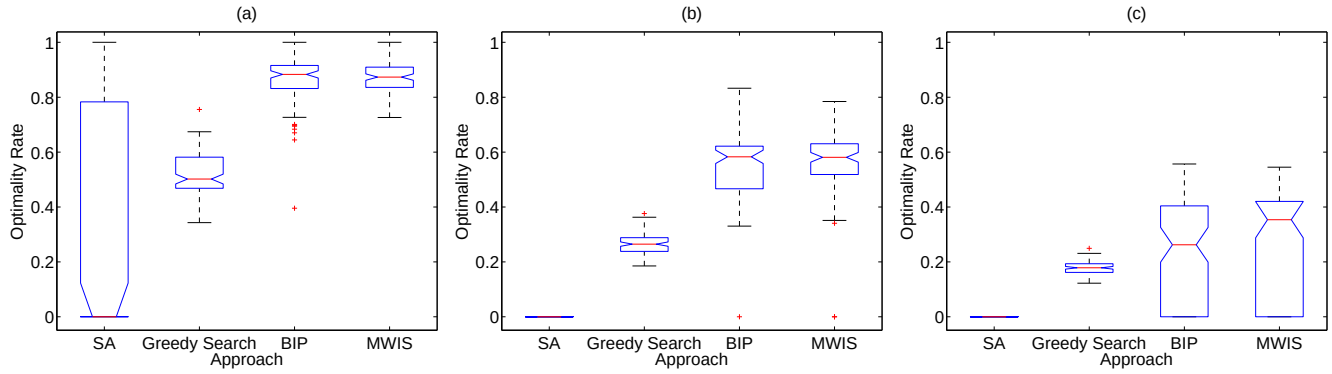


Figura 5. Diagramas de cajas de optimalidad para los diferentes mecanismos: a) 4 agentes, 4 atributos, b) 6 agentes, 6 atributos, c) 8 agentes, 8 atributos.

mediana es cero, lo que significa que al menos la mitad de las veces este enfoque no encuentra solución. Para 4 agentes y 4 atributos (Figura 5.(a)), todos los enfoques propuestos suponen una mejora significativa sobre simulated annealing, aunque la búsqueda voraz tiene claramente una menor tasa de optimalidad que los enfoques basados en selección por torneo. La programación entera y MWIS dan resultados muy similares, ya que ambos realizan la generación de ofertas mediante una maximización de utilidad sobre un subconjunto de restricciones escogido en función del factor Q. Ambos enfoques consiguen tasas de optimalidad medianas cercanas a 0.9. Todos los mecanismos propuestos reducen la tasa de fallos a cero (no hay resultados de optimalidad cero), lo que es un hecho significativo si tenemos en cuenta que la tasa de fallo para simulated annealing está por encima del 50 %. Al aumentar el número de agentes y de atributos (Figura 5.(b) y (c)), la tasa de fallos tanto de la programación entera como de MWIS se incrementan. Por contra, la búsqueda voraz mantiene su tasa de fallo nula. Las optimalidades medianas de todos los enfoques propuestos decrecen al aumentar el número de agentes y atributos, pero siempre manteniéndose por encima del mecanismo de referencia. De estos resultados podemos concluir que el factor de calidad Q puede emplearse para mejorar la tasa de optimalidad y la tasa de fallos en espacios de utilidad altamente no lineales, y que la selección por torneo es una forma factible de seleccionar qué restricciones usar para la generación de pujas. La búsqueda probabilística, aunque alcanza tasas de optimalidad más bajas que el resto de mecanismos, puede ser la mejor opción para escenarios donde se requieran tasas de fallo muy bajas (o incluso nulas) para un número elevado de agentes y atributos.

Los efectos de usar el protocolo expresivo con ciclos de relajación pueden verse en los Cuadros I y II, que muestran los resultados de los experimentos empleando MWIS y el protocolo de negociación expresivo. Para 8 agentes y 8 atributos, puede verse que hay una mejora significativa de la tasa de optimalidad en la segunda iteración, y que sucesivas iteraciones producen ligeros incrementos de la tasa de optimalidad y mejoran de forma significativa la tasa de fallo. Para el caso de 10 agentes y 10 atributos puede verse que, aunque utilizando el protocolo inexpresivo (1 iteración) casi todas las negociaciones fallan, la tasa de fallo se reduce de forma significativa en iteraciones sucesivas. De estos resultados podemos concluir que el protocolo expresivo puede

Cuadro I
PROTOCOLO EXPRESIVO PARA 8 AGENTES Y 8 ATRIBUTOS

	Tasa de optimalidad		Tasa de fallo
# de iteraciones	mediana	int. confianza	%
1	0,3653	[0,2988, 0,4238]	27 %
2	0,4249	[0,4069, 0,4429]	14 %
4	0,4399	[0,4246, 0,4551]	7 %

Cuadro II
PROTOCOLO EXPRESIVO PARA 10 AGENTES Y 10 ATRIBUTOS

	Tasa de optimalidad		Tasa de fallo
# de iteraciones	mediana	int. confianza	%
1	0,0000	[0,0000, 0,0000]	78 %
3	0,0000	[0,0000, 0,0442]	53 %
5	0,2519	[0,2039, 0,3000]	43 %
7	0,2806	[0,2310, 0,3160]	32 %

usarse para mejorar la tasa de optimalidad, y especialmente para mejorar la tasa de fallo en negociaciones en espacios altamente no lineales.

Por lo que respecta al rendimiento, el Cuadro III muestra las medianas y los intervalos de confianza al 95 % para la relación entre el tiempo de negociación utilizando los mecanismos propuestos y el tiempo de negociación del mecanismo de referencia descrito en [8]. La variabilidad de los tiempos de negociación en los diferentes escenarios es grande, no sólo porque la complejidad añadida directamente al aumentar el número de agentes o de atributos, sino también porque el tiempo de la fase de identificación de soluciones aumenta con el número de soluciones viables encontradas. Los valores de tiempo mayores, que son los mostrados en la tabla, se obtuvieron para 8 agentes y 8 atributos. Puede verse que MWIS da como resultado tiempos de negociación significativamente menores a los obtenidos empleando simulated annealing, y que la búsqueda voraz da como resultado tiempos significativamente mayores. No obstante, el tiempo máximo obtenido para la búsqueda voraz es inferior a 30 segundos, lo que puede ser aceptable para algunas aplicaciones, especialmente si se necesita una tasa de fallo nula. Los tiempos para programación entera binaria son mucho mayores y, puesto que da resultados similares a los de MWIS en términos de optimalidad y tasa de fallo, no proporciona ninguna ventaja.

VII. DISCUSIÓN Y TRABAJOS RELACIONADOS

El trabajo seminal que abrió el campo para este trabajo es el artículo de Ito et al. sobre negociación multiatributo

Cuadro III
RENDIMIENTO PARA 8 AGENTES Y 8 ATRIBUTOS

	Relación de Tiempos	
Enfoque	mediana	int. confianza
búsqueda voraz	4,404	[4,157, 4,65]
programación entera	82,76	[81,59, 83, 93]
MWIS	0,71	[0,7098, 0,7226]

en espacios de utilidad no lineales [8]. En él se propone un protocolo de un solo disparo basado en subastas que emplea simulated annealing para identificar regiones de alta utilidad en los espacios de utilidad de los agentes para enviarlas como ofertas a un mediador. Hemos usado este trabajo como punto de partida para desarrollar mecanismos de oferta e identificación de acuerdos que sean efectivos en espacios de utilidad altamente no lineales, donde la "estrechez" de las regiones de alta utilidad de los agentes hace que la tasa de fallo de este enfoque se incremente de forma drástica. En lugar de realizar un muestreo directo del espacio de contratos, nuestras propuestas se aprovechan de la estructura de las preferencias de los agentes y emplean diferentes técnicas sobre el espacio de restricciones para generar las ofertas. La programación entera y los conjuntos independientes máximos han sido usados previamente con éxito en subastas combinacionales [6], [4]. En este artículo, combinamos estos enfoques con una selección por torneo [14] para proporcionar mecanismos de oferta efectivos en escenarios altamente no lineales. Esta selección se sesga empleando un factor de calidad Q, que sirve para equilibrar la utilidad de la oferta y la probabilidad de que la oferta resulte en un acuerdo. Este enfoque es en cierto modo similar a la noción de *viabilidad* que encontramos en [11] para negociación basada en restricciones difusas o a los criterios de similaridad que se emplean en [3] para espacios de utilidad lineales.

Otra técnica para abordar la no linealidad en negociación es aproximar las funciones de utilidad por medio de técnicas de regresión lineal o métodos de media ponderada, tal y como se propone en [5]. No obstante, como los propios autores reconocen en su trabajo, estos enfoques no son útiles para espacios altamente no lineales.

Finalmente, existen otros trabajos que sugieren el uso de protocolos de negociación expresivos en negociación multiagente. En [13] se utiliza información de gradiente para sesgar la búsqueda de soluciones en negociaciones lineales sin mediador, y en [12] se emplean requerimientos de relajación en negociaciones bilaterales comprador-vendedor.

VIII. CONCLUSIONES Y TRABAJO FUTURO

La efectividad de las aproximaciones a la negociación basadas en protocolos de subasta existentes para escenarios no lineales disminuye drásticamente cuando nos enfrentamos a escenarios altamente no lineales, donde las regiones de alta utilidad de los agentes son muy "estrechas" y por lo tanto es muy poco probable que las ofertas de alta utilidad se solapen. Este artículo presenta un conjunto de mecanismos de generación de ofertas que equilibran la "anchura" y la utilidad de las mismas, y un protocolo de negociación expresivo que permite a los agentes mejorar de forma progresiva sus ofertas con cada iteración. Los experimentos demuestran que los mecanismos propuestos suponen una mejora significativa sobre las aproximaciones anteriores en escenarios altamente no lineales, tanto en términos de tasa de fallo como de optimalidad. No obstante, el trabajo deja abiertas varias líneas de trabajo futuro. Estamos realizando un análisis del impacto de los parámetros α y β sobre la efectividad de los diferentes mecanismos. Además, estamos interesados en el diseño de diferentes mecanismos de relajación de ofertas además del que aquí se propone. Por último, estamos trabajando en la generalización de estos mecanismos a otros tipos de funciones de utilidad.

AGRADECIMIENTOS

Este trabajo se ha desarrollado parcialmente en el marco del proyecto ITEA-2 2008005, "Do-it-Yourself Smart Experiences", y ha sido parcialmente financiado por el Ministerio de Educación, dentro del proyecto TIN2008-06739-C04-04.

REFERENCIAS

[1] M. Bayati, D. Shah, and M. Sharma. Max-product for maximum weight matching: Convergence, correctness, and lp duality. *IEEE Transactions on Information Theory*, 54(3):1241–1251, 2008.

[2] Y. Chevaleyre, U. Endriss, S. Estivie, and N. Maudet. Multiagent resource allocation with k-additive utility functions. In *Proceedings of the DIMACS-LAMSADE Workshop on Computer Science and Decision Theory*, pages 83–100, 2004.

[3] P. Faratin, C. Sierra, and N. Jennings. Using similarity criteria to make negotiation trade-offs. In *Proceedings of the 4th International Conference on Multi-Agent Systems*, pages 119–126, 2000.

[4] V. U. Gnanasekaran. Improved opportunity cost algorithm for carrier selection in combinatorial auctions. Master's thesis, Louisiana State University, 2004.

[5] K. Hindriks, C. M. Jonker, and D. Tykhonov. Eliminating interdependencies between issues for multi-issue negotiation. In *Cooperative Information Agents X*, volume 4149 of *Lecture Notes in Computer Science*, pages 301–316, 2006.

[6] G. Hohner, J. Rich, E. Ng, G. Reid, A. J. Davenport, J. R. Kalagnanam, H. S. Lee, and C. An. Combinatorial and quantity-discount procurement auctions benefit mars, incorporated and its suppliers. *Interfaces*, 33(1):23–35, 2003.

[7] T. Ito, M. Klein, and H. Hattori. A multi-issue negotiation protocol among agents with nonlinear utility functions. *Multiagent and Grid Systems*, 4(1):67–83.

[8] T. Ito, M. Klein, and H. Hattori. Multi-issue negotiation protocol for agents: Exploring nonlinear utility spaces. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI07)*, pages 1347–1352, 2007.

[9] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Protocols for negotiating complex contracts. *IEEE Intelligent Systems*, 18(6):32–38, 2003.

[10] V. Kumar and L. Kanal. A general branch-and-bound formulations for understanding and synthesizing and/or tree search procedures. *Artificial Intelligence*, 21:179–198, 1983.

[11] M. A. Lopez-Carmona and J. R. Velasco. An expressive approach to fuzzy constraint based agent purchase negotiation. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS06)*, 2006.

[12] M. A. Lopez-Carmona, J. R. Velasco, and I. Marsa-Maestre. The agents' attitudes in fuzzy constraint based automated purchase negotiations. In *Multi-Agent Systems and Applications V*, volume 4696/2007 of *Lecture Notes in Artificial Intelligence*, pages 246–255. Springer, 2007.

[13] I. Marsa-Maestre, M. A. Lopez-Carmona, and J. R. Velasco. Improving trade-offs in bilateral negotiations under complete and incomplete information settings. In *Proceedings of the Pacific Rim International Conference on Multi-Agents*, volume 5357 of *Lecture Notes in Artificial Intelligence*, pages 275–286, 2008.

[14] B. L. Miller and D. E. Goldberg. Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, 9(3):193–212, 1995.

[15] N. Nisan. *Bidding languages for combinatorial auctions*, chapter 9. MIT Press, 2006.

[16] A. Schrijver. *Theory of Linear and Integer Programming 1998*. John Wiley and Sons, 1998.

Reverse OAuth: Una Solución para la Obtención de Single Sign-On en Entornos de E-Learning

Jorge Fontenla González, Manuel Caeiro Rodríguez, Martín Llamas Nistal
Departamento de Ingeniería Telemática
Universidad de Vigo
ETSI Telecomunicación. Campus Universitario s/n. 36310 Vigo (España).
{jfontenla, Manuel.Caeiro, Martin.Llamas}@det.uvigo.es

***Resumen*—El actual ambiente educativo y socio-tecnológico ha llevado a la proliferación de sistemas de e-learning conocidos como *Learning Management Systems*. Estos sistemas consisten en una aplicación central para gestionar la secuenciación de tareas de estudiantes y docentes, y en otro conjunto de aplicaciones educativas que permiten a aquéllos comunicarse, llevar a cabo experimentos, etc. No obstante, a pesar de su uso generalizado estos sistemas presentan problemas de usabilidad cuando ambos tipos de aplicaciones requieren diferentes procesos de autentificación. En efecto, los usuarios tienen que introducir diferentes conjuntos de credenciales, impidiéndoles enfocar sus esfuerzos en sus estudios y aumentando el conocido como "password stress". Algunas iniciativas han tratado con el problema de la concesión de autorizaciones delegadas, pero sus requisitos son diferentes a los que surgen de un entorno de e-learning, y por tanto no son aplicables. En este artículo presentamos Reverse OAuth, un protocolo para permitir la concesión de autorizaciones para acceder a recursos protegidos en entornos de e-learning.**

***Palabras Clave*—Autorización Delegada, E-Learning, Password Stress, Reverse OAuth, Single Sign-On, Web Service**

I. Introducción

El vertiginoso avance científico y tecnológico ha llevado a organizaciones como universidades y empresas a proporcionar formación contínua a sus estudiantes y empleados a través de sistemas de e-learning conocidos como LMSs (*Learning Management Systems*). Los actuales LMSs pueden ser consideradas como aplicaciones Web representativas y complejas. Ejemplos conocidos de LMSs son Moodle [2] o Blackboard [3]. Estos sistemas típicamente ofrecen un entorno centralizado para suministrar datos (documentos pdf, archivos multimedia, etc.) junto con aplicaciones y herramientas para manipularlos. No obstante, la creciente complejidad de los LMSs ha conducido a una aproximación de diseño en la que las herramientas son extraídas del propio LMS e *invocadas* remotamente [5].

Esta aproximación presenta dificultades relacionadas con la usabilidad de estos sistemas, y debido a lo innovador de la aproximación de separar un LMS y sus herramientas todavía no han sido afrontadas. Principalmente sería deseable que los estudiantes no tuviesen que autentificarse en las herramientas dado que previamente ya han sido autentificados en el LMS. Esto puede verse como la contrapartida en los LMS "tradicionales" que contienen las herramientas así como el propio LMS: un estudiante sólamente tiene que autentificarse al principio de la sesión, pero a partir de ese momento puede usar las herramientas libremente sin tener que autentificarse de nuevo. Este principio de autentificación se conoce habitualmente como *single sign-on*.

El estudio de una forma de conseguir single sign-on entre el LMS y las diferentes herramientas Web es el propósito de este artículo. Tomamos como punto de partida tres tecnologías de single sign-on, analizamos su adecuación a nuestro entorno de e-learning y presentamos nuestra propuesta para solventar los problemas identificados.

Este artículo se estructura como sigue. En la Sección II se describe un escenario típico en el que se necesita una tecnología de single sign-on, y enumeramos los requisitos que deben satisfacerse. La Sección III hace una breve descripción de las tres tecnologías de single sign-on, y las analiza según los requisitos de la Sección II. En la Sección IV lanzamos nuestra propuesta de solución para solventar las carencias identificadas en la Sección III, y en la Sección V proporcionamos su prueba de concepto. Terminamos el artículo con la Sección VI, donde extraemos algunas conclusiones.

II. Descripción del problema

En la Sección I se mencionaron algunas tecnologías de autorización de propósito general. No obstante, para analizar su adecuación a nuestros propósitos es preciso formalizar el problema. Esta sección ofrece una visión más clara de la arquitectura considerada y las funcionalidades esperadas.

A. Descripción de la arquitectura

La Fig. 1 muestra la arquitectura bajo estudio. En ella se aprecian tres entidades: el LMS, la herramienta y el usuario:

- Por un lado, el LMS proporciona el núcleo de funcionalidad del sistema (e.g. bases de datos de usuarios, la lógica para manejar la secuenciación de tareas).
- Por otro lado, la herramienta Web es una aplicación autónoma que proporciona una funcionalidad específica que puede ayudar a los usuarios a llevar a cabo sus tareas (e.g. un foro, un simulador de mecánica de fluídos).
- Finalmente el usuario accede al LMS para llevar a cabo sus tareas con la ayuda de la herramienta Web.

Este desacoplamiento permite a los LMSs extender sus funcionalidades fácilmente, pues para ello únicamente deben establecer una conexión con la herramienta Web apropiada.

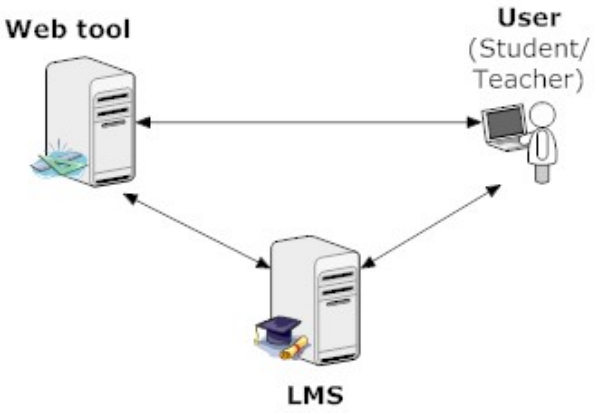


Fig. 1. Representación de la arquitectura.

Los desarrolladores del LMS y de las herramientas Web deben adoptar ciertas especificaciones para soportar su interacción mútua, representada en la Fig. 1 mediante una flecha que une el LMS con la herramienta. Ello permite a los usuarios llevar a cabo las tareas indicadas por el LMS usando la herramienta Web, representado en la Fig. 1 por las flechas que parten del usuario.

La herramienta Web posiblemente tenga implementado algún tipo de control de acceso para prevenir su utilización por parte de usuarios no autorizados. En este documento consideramos básicamente dos posibilidades:

1) Cada usuario tiene cuenta de trabajo en la herramienta.
2) El LMS tiene cuenta de trabajo en la herramienta, y a sus usuarios les concede acceso como *guest users*. Este mecanismo es conocido como *autorización delegada*.

La primera solución no es ni escalable ni práctica, pues los usuarios deben recordar numerosas contraseñas, lo cual redunda en una mala experiencia de usuario. La segunda solución solventa los inconvenientes de la primera al estar basada en un mecanismo de autorización delegada. Por ello, la hemos escogido como base de nuestra solución.

B. Casos de uso

Para clarificar la funcionalidad esperada del sistema introducimos aquí algunos casos de uso basados en un usuario que quiere llevar a cabo las actividades de una asignatura "Hidrodinámica", impartida desde un LMS alojado y mantenido por la Universidad A. Para ayudar a los usuarios con sus actividades hay disponibles vía Web dos simuladores de fluídos, cada uno con sus propias funcionalidades. Estos simuladores están alojados y mantenidos por la Universidad B. Además, el usuario debe responder a un cuestionario online. Esta funcionalidad la proporciona una herramienta Web de evaluación alojada en la Universidad C. El usuario puede usar cualquiera de los simuladores de la Universidad B para responder a las preguntas planteadas por la herramienta de evaluación. El límite de tiempo del cuestionario es una hora:

1) El LMS proporciona al usuario hiperenlaces a la herramienta de evaluación y a los simuladores. Desde el momento en que sigue el enlace al cuestionario el usuario dispone de una hora para responder a las preguntas. Durante ese periodo puede usar los simuladores según sus necesidades. Cuando la hora ha transcurrido no se le permite seguir respondiendo al cuestionario.
2) Cuando el usuario sigue los enlaces a alguna de las tres herramientas, el LMS establece una negociación con la herramienta en cuestión para conceder al usuario un acceso transparente.
   Mientras el usuario realiza sus actividades el LMS puede recibir informes de las herramientas, relativas a lo que el usuario está haciendo en ese momento (e.g. cuántas veces el usuario ha cambiado la respuesta en una pregunta determinada). Esta información es convenientemente procesada por el LMS.
   Cuando la hora ha transcurrido la calificación es transferida de la herramienta al LMS, donde es almacenada.
3) La herramienta recibe una conexión entrante procedente del LMS destinada a establecer la duración de su uso por el usuario y los ficheros a los que se le permite acceder. Este proceso de negociación no involucra datos sensibles relativos al usuario, sino únicamente parámetros de acceso.
   Una vez estos parámetros han sido negociados, el usuario obtiene acceso a la herramienta. Mientras completa el cuestionario, la herramienta notifica eventos al LMS relativos a sus actividades. Cuando la hora ha transcurrido, o cuando el usuario sale de la herramienta de evaluación, ésta notifica un último evento al LMS concerniente a la calificación final del usuario.

C. Enumeración de requisitos

En base a los casos de uso podemos enumerar formalmente los requisitos de la solución. Un subconjunto de ellos han sido identificados como muy urgentes por algunas comisiones de e-learning [4]:

1) **Interoperabilidad:** el LMS debe poder interoperar con una herramienta Web incluso estando en dominios de red diferentes.
2) **Transparencia de Acceso:** los usuarios deben poder acceder a la herramienta sin que se les solicite autentificación, pues previamente ya se han autentificado ante el LMS. La transparencia de acceso es parte del caso de uso 2.
3) **Privacidad:** la herramienta únicamente puede tener acceso a aquellos datos que le suministre el propio LMS. En el caso de uso 3, las Universidades B y C no tienen acceso a información sensible sobre el estudiante de la Universidad A (e.g. nombre, e-mail).
4) **Elegibilidad:** un usuario del LMS debe poder acceder a la herramienta que prefiera de entre aquellas que le ofrezca el LMS. Este requisito está implícito en el caso de uso 1.
5) **Granularidad:** un usuario debe poder acceder a recursos particulares (e.g. ficheros concretos) en la herramienta con diferentes tipos de permisos (e.g. lectura,

ejecución). En el caso de uso 1, por ejemplo, un usuario sólo tiene acceso a su propia instancia del cuestionario y no a la de otros usuarios.

6) **Simplicidad:** la solución debe ser simple y escalable, y no involucrar a más actores que el LMS, la herramienta y el usuario.
7) **Reconfiguración Dinámica:** el LMS debe poder modificar las características de una autorización en curso. En el caso de uso 2, por ejemplo, el LMS puede desear conceder acceso al usuario a una parte opcional del cuestionario en tiempo de ejecución.
8) **Expiración:** las autorizaciones deben tener un periodo de validez. Así, en nuestro ejemplo la sesión con la herramienta de evaluación debe terminar al cabo de una hora.
9) **Percepción:** el LMS debe poder monitorizar las actividades de cada usuario de la herramienta. En efecto, como se indica en el caso de uso 2, el LMS puede tomar decisiones según la información que recibe de las herramientas.
10) **Pseudonimidad:** dado que una herramienta no debe tener acceso a la identidad del usuario por el Requisito 3 (Privacidad), aquélla debe permitir distinguir a sus usuarios con el fin de diferenciar sus actividades.
11) **Confidencialidad:** los datos sensibles enviados entre las entidades deben ser mantenidos como confidenciales ante ataques de escucha. Por ejemplo, la información con la que el LMS concede acceso a la herramienta de evaluación en el caso de uso 2 debe ser confidencial, pues un atacante podría interceptarla y reutilizarla para obtener acceso ilimitado a la herramienta.
12) **Integridad:** debe ser posible detectar modificaciones ilícitas enviadas entre el usuario, el LMS y la herramienta. Es importante, por ejemplo, que el tiempo de acceso a la herramienta de evaluación no pueda ser ser cambiado ilícitamente.
13) **Autenticidad:** el mecanismo de autorizaciones delegadas debe detectar si alguna de las entidades ha sido suplantada. Esto es importante en actividades educativas como el cuestionario de nuestro ejemplo.
14) **Autorizaciones de Uso Único:** el usuario no puede reutilizar autorizaciones ya expiradas para intentar acceder de nuevo al recurso protegido. En nuestro ejemplo el usuario no debe poder reintentar el examen usando la autorización que originalmente le fue concedida.

III. Trabajos relacionados

Desde la aparición de la Web 2.0 muchos sitios Web pueden interoperar para alcanzar funcionalidades nuevas. Un ejemplo de esto es NetVibes [6], que permite recolectar información de otros sitios de Internet en los que el usuario tiene cuenta (e.g. Gmail, Yahoo! Mail) para mostrarla toda junta. No obstante, estos sitios exhiben un importante problema de privacidad. NetVibes, por ejemplo, puede solicitar al usuario la contraseña de su cuenta de e-mail. Desde ese momento juega un importante lugar la confianza que tiene el usuario en que NetVibes no haga un uso ilícito de la contraseña (e.g. leer correos privados, enviar correos suplantando al usuario, cambiar la contraseña).

El caso de NetVibes es bastante frecuente. En vista de este escenario han aparecido algunas soluciones de single sign-on, de forma que un usuario pueda acceder a varios sistemas software autentificándose una única vez. Entre los beneficios del single sign-on destacan la reducción del tiempo necesario para reintroducir las contraseñas por parte del mismo usuario, y la reducción del esfuerzo necesario para recordar múltiples contraseñas, conocido como *password stress*.

En nuestro contexto los usuarios son estudiantes y profesores, y los sistemas que requieren autentificación son el LMS y las herramientas Web. El single sign-on es útil en este escenario, pues los usuarios ya se han autentificado ante el LMS y (desde su punto de vista) autentificaciones adicionales ante cada herramienta no deberían ser necesarias.

Entre las diferentes formas de conseguir single sign-on hemos escogido tres que son bastante relevantes para nuestros propósitos: OAuth [7], Delegation Permits [8] y Shibboleth [9]. Estas tecnologías han sido propuestas para conceder autorizaciones delegadas en escenarios single sign-on involucrando aplicaciones Web de propósito general. Tecnologías emergentes como OpenID intentan dar solución al problema de *autentificaciones* delegadas, y por tanto caen fuera del ámbito del presente artículo. Nuestro problema a tratar son las *autorizaciones* delegadas.

A. OAuth

Mediante OAuth [7] un usuario puede conceder tickets a una aplicación Web para acceder a recursos protegidos alojados en otra ubicación, sin tener que confiarle ningún conjunto de credenciales. Estos recursos protegidos pueden ser datos (e.g. imágenes, documentos), acciones (e.g. crear un nuevo hilo en un foro, enviar un e-mail) o, en general, cualquier URL con restricciones de acceso.

En la página oficial de OAuth puede encontrarse mucha información sobre el protocolo, por lo que aquí únicamente haremos un breve resumen. Su funcionamiento, ilustrado en la Fig. 2, tiene lugar en una arquitectura compuesta de tres actores principales:

- ***Service Provider***: aplicación Web que permite el acceso a recursos protegidos vía OAuth.
- ***User***: persona con cuenta de trabajo en el Service Provider.
- ***Consumer***: aplicación que emplea OAuth para acceder al Service Provider en nombre del User.

El funcionamiento interno de OAuth está basado en el uso de tickets de un solo uso, o "tokens" empleando la terminología de OAuth. Para que estas tokens sean válidos para acceder al recurso deben ser previamente autorizados por el User. OAuth diferencia dos tipos de tokens. Las Access Tokens permiten a su poseedor acceder al recurso protegido. Las Request Tokens, por su parte, permiten a su poseedor obtener una Access Token.

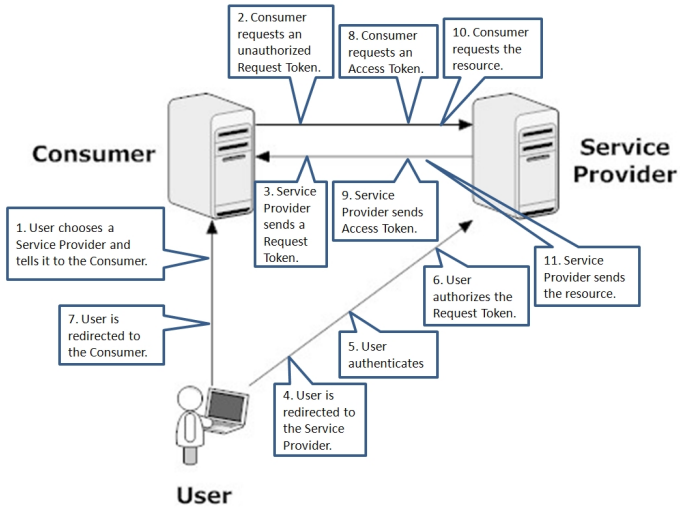


Fig. 2. Intercambio de mensajes en OAuth.

Una peculiaridad de OAuth es su aproximación "todo o nada". OAuth puede ser usado para conceder acceso sin restricciones a todos los recursos alojados en el Service Provider. Éste último puede restringir los privilegios de acceso del Consumer por iniciativa propia (e.g. permitiendo acceso de sólo lectura a determinados recursos), pero ya no es parte del propio OAuth.

B. Delegation Permits

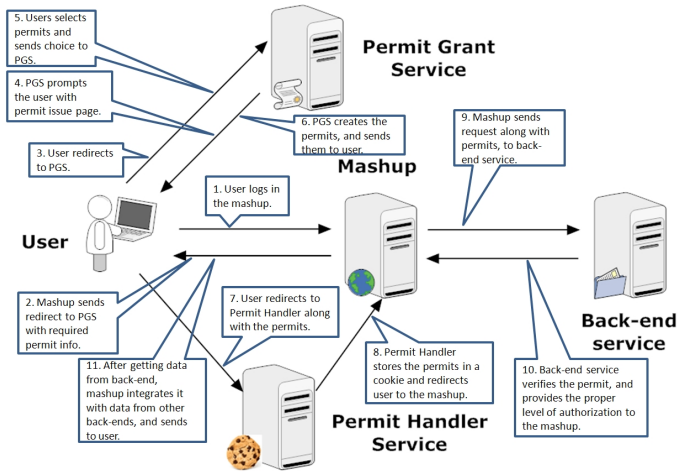


Fig. 3. Intercambio de mensajes en Delegation Permits (figura adaptada de [8]).

Delegation Permits [8] (en adelante DP) trata con el problema de conceder autorizaciones a mashups para acceder a recursos protegidos, aunque su funcionamiento puede ser generalizado a otro tipo de sistemas. Básicamente, DP funciona sobre la arquitectura mostrada en la Fig. 3, en la que se pueden apreciar cinco entidades:

- ***Mashup*:** aplicación que quiere acceder al Back-end Service.
- ***User*:** usuario del Mashup y del Back-end Service.
- ***Back-end Service*:** aplicación que aloja los datos a los que el Mashup quiere acceder.
- ***Permit Grant Service*:** aplicación que expide autorizaciones para acceder al Back-end Service.
- ***Permit Handler Service*:** aplicación ejecutándose en el Mashup, responsable de la gestión de autorizaciones una vez son recibidas del Permit Grant Service.

La principal diferencia arquitectural con OAuth reside en que en DP hay más entidades involucradas: el Permit Handler Service y el Permit Grant Service. Mientras que el Permit Handler Service sólo proporciona una optimización del rendimiento del protocolo mediante el uso de cookies (y por tanto es una entidad prescindible), en el Permit Grant Service reside el núcleo de funcionalidad del protocolo. Proporciona un entorno centralizado donde el User puede conceder o denegar la autorización al Mashup. Así, el Permit Grant Service es un punto único de fallo, y que además puede suponer un potencial cuello de botella.

C. Shibboleth

La función principal de Shibboleth [9] es soportar el acceso transparente a recursos en múltiples sitios Web entre los que existe una relación de confianza conocida como *federación*. El funcionamiento de Shibboleth está basado en SAML 2.0, un dialecto de XML para intercambiar información de seguridad entre dominios federados. La información es codificada como parejas clave–valor conocidas como *atributos* (e.g. "branch–telematics").

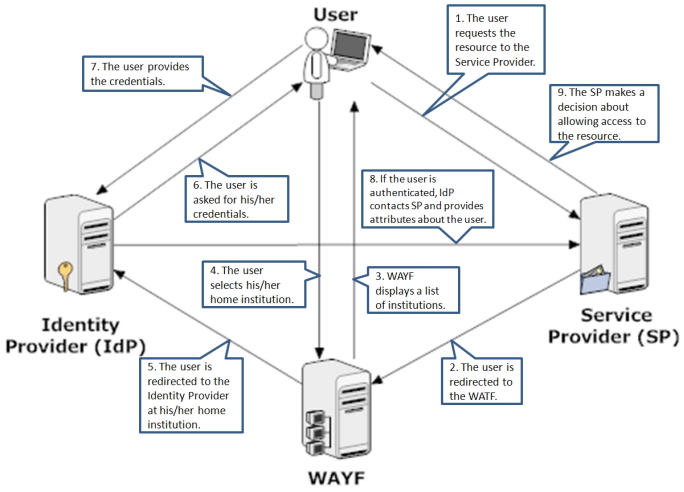


Fig. 4. Intercambio de mensajes en Shibboleth.

Shibboleth opera en una arquitectura compuesta por cuatro actores (ver Fig. 4):

- ***Identity Provider*:** proceso ejecutándose en la institución principal del User, responsable de autentificarlo ante toda la federación.
- ***Service Provider*:** institución de la federación que almacena el recurso protegido al que el User quiere acceder.

- ***User*:** persona con cuenta de usuario en el Identity Provider que quiere acceder al recurso protegido alojado en el Service Provider.
- ***WAYF (Where Are You From)*:** servicio centralizado de la federación que permite al User elegir su institución principal.

Cuando el User quiere acceder al recurso protegido navega a su URL usando su navegador. El Service Provider redirige al navegador al WAYF, donde el User puede escoger su organización principal. El navegador es enviado a la página de la organización principal, donde introduce sus credenciales. El Identity Provider envía al navegador de nuevo al Service Provider avisándole de si ha sido correctamente identificado.

El proceso es completo cuando el Service Provider solicita atributos adicionales sobre el User al Identity Provider. Estos atributos no involucran necesariamente la identidad del usuario, aunque también podría ser solicitada. El Service Provider usa estos atributos para decidir la política de acceso para el User.

Una carencia importante de Shibboleth es la del proceso complementario al single sign-on, conocido como *single log-out*. Actualmente no sólo no está soportado por Shibboleth, sino que sus requisitos ni siquiera están claros. La ausencia de single log-out tiene la importante consecuencia de que no es fácil saber si el User todavía está logueado en la federación, o si su sesión ha terminado.

D. Análisis comparativo

Aunque está claro por las secciones predecentes que las tecnologías anteriores tratan con el problema de la concesión de autorizaciones para acceder a sistemas de terceros, su ámbito es ligeramente diferente del nuestro. La Tabla I muestra una comparación de estas tecnologías según nuestros requisitos. De acuerdo con esta tabla es importante darse cuenta de las siguientes cuestiones:

- La ausencia de single log-out en Shibboleth implica que no es fácil saber si el usuario está todavía accediendo a la federación o no, y por ello debe autentificarse de nuevo cada vez que quiera acceder a un recurso protegido. En consecuencia Shibboleth no satisface el Requisito 2 (Transparencia de Acceso). OAuth y DP satisfacen el requisito pues no requieren procesos de autentificación adicionales.
- Los atributos concretos que son enviados en Shibboleth no están limitados por la especificación sino que son negociados por la federación. Así, es posible que los atributos enviados se refiriesen a datos personales de los usuarios. Así, el cumplimiento del Requisito 3 (Privacidad) no está garantizado. OAuth y DP, por su parte, no necesitan información sensible para operar y por tanto cumplen el requisito.
- El aspecto más importante es que las diferencias entre nuestra arquitectura (Sección II-A) y aquellas asumidas en OAuth y DP descartan automáticamente el uso directo de estas tecnologías. En efecto, en estas tecnologías una persona autoriza a un servicio para acceder a otro servicio. En nuestro caso, sin embargo, queremos que un servicio autorice a una persona para acceder a otro servicio. Este hecho tiene la importante implicación de que tanto en OAuth como en Delegation Permits la persona no tiene libertad para escoger la herramienta Web. Así, OAuth y DP incumplen el Requisito 4 (Elegibilidad).
  Por su parte Shibboleth encaja mejor en nuestro modelo, pues ha sido pensado para permitir a una persona acceder a recursos de una federación. Por ello Shibboleth, a diferencia de OAuth y DP, cumple el requisito.
- OAuth no proporciona medios para acceder a recursos protegidos específicos, sino que en su lugar sigue una aproximación "todo o nada" por la cual un Consumer puede obtener acceso a todos los recursos alojados en el Service Provider. Aunque esta aproximación puede ser interesante en algunos escenarios, en el nuestro necesitamos que los usuarios puedan acceder a recursos específicos. Por otro lado DP y Shibboleth sí consideran el acceso a recursos específicos con permisos específicos. Es por ello que cumplen el Requisito 5 (Granularidad) pero no así OAuth.
- DP no cumple el Requisito 6 (Simplicidad) pues involucra a cinco entidades en lugar de sólo al usuario, la herramienta y la herramienta. Además, el Permit Grant Service es un servicio complejo que alberga el núcleo de funcionalidad de DP y que por tanto supone un cuello de botella potencial y un punto único de fallo. De forma similar, el correcto de una federación en Shibboleth depende de un único servicio WAYF. Por el contrario, OAuth se basa en una arquitectura muy simple que sólo involucra a estas tres entidades, y cuya carga computacional está bien equilibrada entre todas ellas.
- El Requisito 7 (Reconfiguración Dinámica) no está soportado por ninguno de los protocolos bajo estudio. La razón es que no han sido diseñados para escenarios en que las autorizaciones están sujetas a planificaciones susceptibles de cambio, que es el caso en entornos de e-learning. Shibboleth es el que proporciona mayor nivel de versatilidad al configurar las autorizaciones; no obstante el proceso de configuración únicamente tiene lugar al principio cuando la autorización está siendo negociada, sin posibilidad de cambiar las propiedades de una autorización en curso.
- DP incluye como parámetro el instante de expiración de la autorización. De forma similar, el tiempo de expiración puede ser enviado como atributo en Shibboleth. No obstante, en el caso de OAuth el tiempo de expiración es impuesto unilateralmente por el Service Provider y no puede ser establecido o controlado por el User. Por tanto, DP y Shibboleth satisfacen el Requisito 8 (Expiración) pero no así OAuth.
- Dado que OAuth y DP están basados en un modelo de negocio diferente, la provisión de feedback desde la herramienta Web al LMS ni siquiera ha sido considerada. Por su parte Shibboleth, aunque plantea un modelo semejante, no considera ninguna comunicación entre el Identity Provider y los Service Provider tras la fase inicial

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14
OAuth	✓	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	✓	✓	✓
DP	✓	✓	✓	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✓
Shibboleth	✓	✗	✗	✓	✓	✗	✗	✓	✗	✓	✓	✓	✓	✓

Tabla I
COMPARACIÓN ENTRE OAUTH, DP Y SHIBBOLETH.

de negociación, lo que excluye cualquier posibilidad de enviar feedback. Consecuentemente OAuth, DP y Shibboleth incumplen el Requisito 9 (Percepción).

- Ni OAuth ni DP posibilitan que la herramienta pueda identificar anónimamente a los estudiantes. Por tanto, no cumplen el Requisito 10 (Pseudonimidad). Shibboleth, por su parte, puede soportar pseudónimos si son enviados como atributos, supuesto que han sido acordados por la federación como atributos aceptados.
- Finalmente, a diferencia de OAuth que proporciona mecanismos para verificar la integridad de los mensajes y la autenticidad de los actores involucrados (Requisitos 12 y 13), DP no tiene en cuenta estos aspectos. En cualquier caso, ninguno de los dos protocolos proporciona confidencialidad (Requisito 11). Respecto a Shibboleth, satisface los Requisitos 11, 12 y 13. No obstante lo hace mediante el uso de SSL, y por tanto una comparación justa con OAuth y DP no es posible.

Vemos que estas tecnologías exhiben algunas características deseables, pero que han sido diseñadas para escenarios diferentes y por tanto no satisfacen todos nuestros requisitos. La posibilidad más factible es realizar un rediseño de alguna de estas tecnologías para aprovecharse de sus ventajas actuales pero solventando sus limitaciones.

Para ello tomamos OAuth como punto de partida por su carácter abierto y la fuerte comunidad de desarrollo tras él, y por su simplicidad arquitectural. El resultado es una variante de OAuth que hemos dado en llamar Reverse OAuth, pues es el usuario (Consumer) quien indica al LMS (User) qué herramienta usar, al contrario de lo que ocurre en la versión normal de OAuth. El resto de este artículo proporciona una descripción en profundidad de Reverse OAuth.

IV. UNA SOLUCIÓN SINGLE SIGN-ON PARA SISTEMAS DE E-LEARNING. REVERSE OAUTH

Con Reverse OAuth solventamos las carencias mencionadas en la Sección III-D. Este protocolo está fuertemente basado en OAuth, pero incluyendo algunas características de DP y Shibboleth. En esta sección describimos el funcionamiento del protocolo.

A. Descripción general

Emplearemos la misma terminología de OAuth para enfatizar las similitudes de ambos protocolos. Así, tenemos:

- ***Service Provider***: aplicación Web que aloja aquellos recursos protegidos que son accedidos vía Reverse OAuth. En nuestro escenario, el rol de Service Provider es desempeñado por las herramientas Web.
- ***User***: aplicación con cuenta en el Service Provider. En nuestro caso es el LMS quien desempeña este papel, teniendo cuentas en todas aquellas herramientas Web que ofrezca a sus usuarios.
- ***Consumer***: persona que usa Reverse OAuth para acceder al Service Provider en nombre del User, típicamente un usuario del LMS.

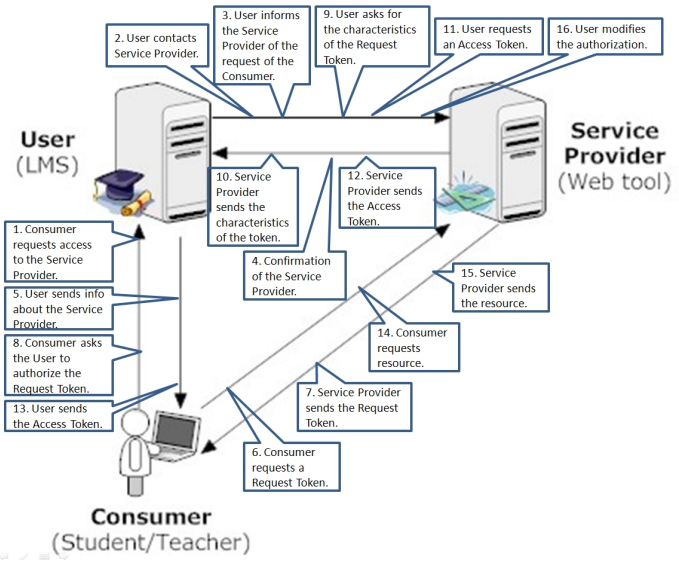


Fig. 5. Intercambio de mensajes en Reverse OAuth.

La Fig. 5 describe la interacción que tiene lugar entre estas tres entidades cuando el Consumer quiere acceder a un recurso protegido alojado en el Service Provider. La secuencia de mensajes que estas entidades se envían es como sigue:

1) Primeramente, el Consumer solicita al User acceder a un recurso protegido alojado en el Service Provider.
2) El User contacta con el Service Provider y se autentifica.
3) El User informa al Service Provider de que el Consumer va a solicitar una Request Token. Con este mensaje el Service Provider puede asociar a un Consumer con el User que concede la autorización.
4) El Service Provider envía una confirmación.
5) El User envía al Consumer información relativa al Service Provider y el recurso deseado. Se incluye la URL en la que el Consumer debe realizar la solicitud, la URL del recurso, las modalidades de acceso (e.g. lectura, escritura, ejecución), la fecha de expiración de la autorización, y una cadena alfanumérica con la que el Service Provider identifica al Consumer.

6) El Consumer contacta con el Service Provider y solicita una Request Token. El mensaje incluye aquellos parámetros que el User envió al Consumer en el paso previo.
7) El Service Provider envía la Request Token al Consumer.
8) El Consumer pide al User que autorice la Request Token y que la cambie por una Access Token.
9) El User pregunta al Service Provider por las características de la Request Token. De esta forma el User pueda comprobar si las características de la Request Token que el Consumer solicitó en el paso 6 son las mismas que las de el paso 5. Si no se llevase a cabo esta comprobación sería posible que el Consumer alterase ilícitamente las características de la Request Token.
10) El Service Provider envía al User las características de la Request Token. El User lleva a cabo la comprobación descrita en el paso anterior. En caso de que no fuese satisfactoria el protocolo se detiene, lo que implcitamente deniega al Consumer el acceso al recurso protegido.
11) El User solicita una Access Token al Service Provider.
12) El Service Provider envía una Access Token al User.
13) El User envía la Access Token al Consumer.
14) El Consumer presenta la Access Token al Service Provider.
15) El Service Provider comprueba la validez de la Access Token. Si es válida, envía el recurso.
16) Si fuese necesario, el User modifica las características de la autorización que ha sido concedida al Consumer.

Todos los mensajes que intervienen en Reverse OAuth son mensajes HTTP, cuyos parámetros están codificados siguiendo el método POST. El lector interesado puede encontrar una descripción del contenido de los mensajes en [1].

B. Cumplimiento de los requisitos con Reverse OAuth

Las peculiaridades de Reverse OAuth hacen que tenga un mejor comportamiento ante nuestros requisitos, ver Tabla II. No obstante, todas ellas surgen del hecho de que en Reverse OAuth el Consumer es una persona y el User un sistema software. A partir de ésta se siguen el resto de diferencias:

- En Reverse OAuth el Service Provider es escogido por el Consumer (el alumno/profesor) y no por el User (el LMS). El Consumer tiene la posibilidad de elegir el Service Provider según sus necesidades. En otras palabras, Reverse OAuth cumple el Requisito 4 (Elegibilidad).
- Reverse OAuth permite especificar el recurso concreto al que se pretende dar acceso al usuario (ver [1]). Por tanto, se cumple el Requisito 5 (Granularidad).
- El diseño arquitectural de Reverse OAuth mantiene la simplicidad de OAuth, involucrando sólamente tres actores. Consecuentemente, Reverse OAuth cumple el Requisito 6 (Simplicidad).
- En Reverse OAuth el User puede modificar las características de la sesión del Consumer en el Service Provider en tiempo de ejecución (paso 16 del protocolo) para adaptarla a las necesidades actuales. Por tanto, cumple el Requisito 7 (Reconfiguración Dinámica).
- Reverse OAuth, como DP, permite especificar el periodo de validez de una autorización (ver [1]). Por tanto, se cumple el Requisito 8 (Expiración).
- Reverse OAuth incluye información en sus mensajes (ver [1]) que pueden ser usados por el User para monitorizar las actividades del Consumer en el Service Provider. Esto implica el cumplimiento del Requisito 9 (Percepción).
- En el paso 3 el alumno/profesor se identifica ante la herramienta mediante un pseudónimo (ver [1]). Este pseudónimo es usado durante el resto del protocolo en lugar de su verdadera identidad. Por tanto, Reverse OAuth cumple el Requisito 10 (Pseudonimidad).
- El uso de firmas en los mensajes de Reverse OAuth (ver [1]) permite detectar tanto modificaciones en los mensajes como suplantación de los actores involucrados en el protocolo. En otras palabras, se cumplen los Requisitos 12 y 13.
- Una vez la autorización ha expirado (o el User la ha revocado) las correspondientes Request Token y Access Token dejan de ser válidas, lo que asegura que las autorizaciones son de un solo uso. Así, Reverse OAuth cumple el Requisito 14 (Autorizaciones de Uso Único).

Hay que mencionar que Reverse OAuth no incorpora mecanismos para asegurar la confidencialidad de los mensajes intercambiados (incumplimiento del Requisito 11). La razón es que la provisión de confidencialidad requeriría una arquitectura compleja para la distribución de claves y certificados, lo que incrementaría sustancialmente la complejidad del protocolo. Por tanto, hemos decidido considerar un protocolo agnóstico respecto a tecnologías de confidencialidad. No obstante, Reverse OAuth puede ser usado en combinación con otras tecnologías como TLS si fuese necesario.

V. Prueba de concepto

Para demostrar la viabilidad de nuestra aproximación hemos desarrollado un prototipo en nuestra red corporativa consistente en una aplicación Web ya existente llamada SimulNet [10], usada en la Universidad de Vigo en la asignatura *Fundamentos de Ordenadores 1*. SimulNet proporciona un entorno de laboratorio virtual en el que los estudiantes pueden poner en práctica sus conocimientos teóricos sobre programación de bajo nivel.

Hemos adaptado SimulNet para soportar un mecanismo básico de autentificación basado en login–password que permite acceso a la totalidad de funcionalidades de la herramienta. Las credenciales son enviadas como texto en claro. Asimismo hemos creado una cuenta de trabajo en SimulNet, que fue asignada al LMS.

Como LMS desplegamos una instancia de Moodle en una subred diferente a la de SimulNet. Cuando un usuario dirige su navegador a la URL en la que se encuentra Moodle, se le muestra una página Web en la que se le pide que se autentifique. Tras esta fase de autentificación, el usuario accede a la página principal del curso. Hemos creado una materia llamada “Computer Architecture”, y una tarea en esta asignatura llamada “Practice 1”. Cuando el usuario va

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14
OAuth	✓	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	✓	✓	✓
DP	✓	✓	✓	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✓
Shibboleth	✓	✗	✗	✓	✓	✗	✗	✓	✗	✓	✓	✓	✓	✓
R. OAuth	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓

Tabla II
REQUISITOS SATISFECHOS POR REVERSE OAUTH.

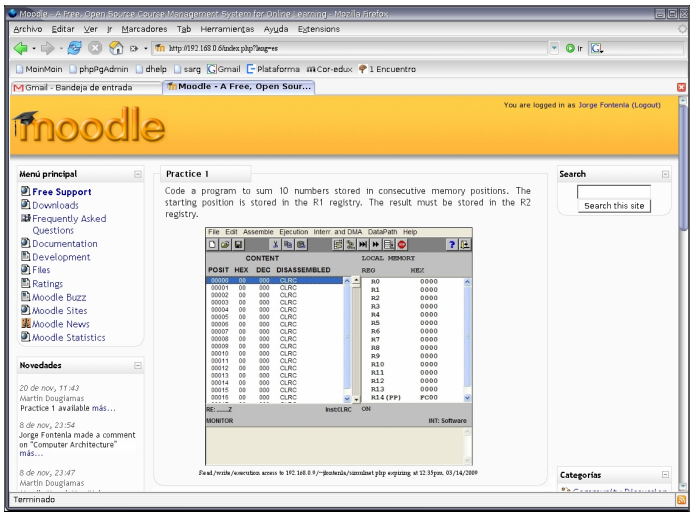


Fig. 6. Acceso a SimulNet vía Reverse OAuth.

a la página de la práctica se le proporciona un hiperenlace a SimulNet. En el momento en que el estudiante hace click en el enlace comienza el proceso descrito en la Sección IV. El resultado se muestra en la Fig. 6.

La usabilidad del diseño ha sido evaluada por algunos miembros del personal de la Universidad de Vigo, vinculados al campo del diseño software y al e-learning. Todos ellos encontraron el prototipo usable y apropiado.

VI. CONCLUSIONES

Los LMSs actuales están jugando un papel importante en la enseñanza, al dar acceso a recursos educativos evitando barreras espaciales y temporales. No obstante, sus posibilidades están limitadas por su creciente complejidad y el problema "una talla no sirve a todos". Estas limitaciones sugieren separar a los propios LMS de las herramientas educativas. Esta aproximación no sólo implica el diseño de nuevos sistemas de e-learning, sino un modelo de negocio completamente nuevo en el que el desarrollo de los LMSs y de las herramientas educativas siguen cursos separados (pero complementarios). La necesidad de single sign-on en este escenario ha sido nuestro punto de partida. Reverse OAuth proporciona una solución ha este problema por medio de un mecanismo de autorizaciones delegadas.

Uno de los objetivos de nuestro trabajo ha sido el desarrollo de un protocolo liviano, abierto e interoperable. Reverse OAuth es agnóstico respecto a lenguajes de programación pues está basado en un protocolo "neutral" como es HTTP. Por tanto, nuestra implementación de Reverse OAuth en Moodle puede ser fácilmente portada a otros LMSs.

Reverse OAuth no es exclusivo del campo del e-learning. Son concebibles otros escenarios en los que la concesión de autorizaciones delegadas es una técnica útil, pero Reverse OAuth ha sido diseñado con requisitos específicos del e-learning en mente. No obstante, es un protocolo abierto y que puede ser modificado a conveniencia. El propio Reverse OAuth es un ejemplo de adaptación de un protocolo previo (OAuth) a un campo con requisitos específicos.

Creemos que Reverse OAuth es un modelo útil para conseguir single sign-on en sistemas de e-learning y que, dada la proliferación de los LMSs esperamos su progresiva adopción por la comunidad investigadora.

VII. AGRADECIMIENTOS

Este trabajo ha sido financiado por el Ministerio de Educación y Ciencia bajo la subvención TIN2007-68125-C02-02, y por la Consellería de Innovación e Industria bajo la subvención PGIDIT06PXIB32 2270PR.

REFERENCIAS

[1] J. Fontenla, M. Caeiro, M. Llamas, L. Anido, "Reverse OAuth - A solution to achieve delegated authorizations in single sign-on environments", Computers and Security. Pendiente de publicación. Accesible en http://dx.doi.org/10.1016/j.cose.2009.06.002

[2] Sitio Web del proyecto Moodle. Accedido en Junio de 2009 en http://moodle.org/

[3] Sitio Web del proyecto Blackboard. Accedido en Junio de 2009 en http://www.blackboard.com/us/index.bbb

[4] T. Klobucar, "Requirements collection and analysis with focus on privacy and security issues", *Prolearn: European Commission Sixth Framework Project (IST-507310)*. Accedido en Junio de 2009 en http://isotc.iso.org/livelink/livelink?func=ll&objId=5906025&objAction=browse

[5] Especificación IMS Tools Interoperability. Accedido en Junio de 2009 en http://www.imásglobal.org/ti/index.html

[6] Sitio Web de NetVibes. Accedido en Junio de 2009 en http://www.netvibes.com/

[7] Especificación OAuth. Accedido en Junio de 2009 en http://oauth.net/core/1.0/

[8] R. Hasan, R. Conlan, B. Slesinsky, N. Ramani, M. Winslett, "Please Permit Me: Stateless Delegated Authorization in Mashups", *Annual Computer Security Applications Conference (ACSAC) 2008*.

[9] Shibboleth Web site. Last accessed on June, 2009 at http://shibboleth.internet2.edu/

[10] L. Anido, M. Llamas, M. Caeiro, J. Santos, J. Rodriguez, M. J. Fernandez, "An update on the SimulNet educational platform. Towards standards-driven e-learning", *IEEE Transactions on Education*, Vol. 44, No. 2, 2001.

Búsquedas epidémicas basadas en perfiles en redes P2P no estructuradas

Juan Vera del Campo, Juan Hernández Serrano, Josep Pegueroles
Departament d'Enginyeria Telemàtica,
Universitat Politècnica de Catalunya
Jordi Girona 1-3, 08034 Barcelona
{juanvi, jserrano, josep}@entel.upc.es

***Resumen*—The success and intensive use of P2P networks has made enhanced strategies for resource searching a hot topic of research. Current solutions are not scalable to several million nodes or centralized; they need complex rather than dynamic structures and do not allow complex queries to be executed. In this paper, we model several mechanisms for data searches based on the interests of the user. We propose additional enhancements on current epidemic algorithms based on the creation of affinity networks. We use simulations to prove that our proposal is comparable to others that are currently deployed and that it solves some of the problems caused by existing solutions.**

***Palabras Clave*—unstructured p2p, clustering, searching, simulation, semantic searches, affinity network**

I. Introducción

Uno de los problemas centrales en redes P2P es cómo los usuarios encuentran los documentos que desean obtener de la red. Las redes P2P más comunes tienen mecanismos de búsqueda centralizados, no escalables o débiles ante ataques legales o técnicos. En muchos casos estos servicios de búsqueda no aceptan peticiones complejas y obligan al usuario a comprender y utilizar con habilidad el mecanismo de búsqueda para obtener resultados especiales.

Con el fin de proveer de búsquedas complejas en redes P2P, surgieron los esquemas de **recomendación** de documentos entre los usuarios de la red. El mayor problema abierto en este caso es cómo filtrar las recomendaciones según la utilidad que tienen para cada usuario. Las propuestas actuales, como veremos, se basan en crear relaciones de amistad ente los usuarios y utilizar búsquedas epidémicas dentro de la red social.

En este artículo proponemos un sistema de búsqueda basado en la creación rápida de una estructura de red según las afinidades de los usuarios. Estas afinidades se calculan dinámicamente por mecanismos provistos por la propia red, y se aprovechan en la etapa de búsqueda de un recurso mediante algoritmos epidémicos. Definiremos a los usuarios, los documentos de la red y las peticiones como perfiles matemáticos de un cierto espacio social, permitiendo así la comparación entre usuarios *a priori* desconocidos, recursos y peticiones. Este espacio social nos permitirá búsquedas semánticas por afinidades de los usuarios, lo que constituye un sistema de recomendación de documentos.

El artículo se estructura de la siguiente manera. La sección II lista los trabajos relacionados con nuestro trabajo. La sección III explora los modelos sociales y de red que usaremos en este documento, e introduce algunas definiciones básicas de nuestro problema. En la sección IV exploramos el escenario de estudio y la formalización del objetivo a alcanzar. La sección V presenta un modelo que permite el estudio de las ideas tradicionales de búsquedas epidémicas en el escenario social descrito, y simula los resultados de este modelo. En la sección VI proponemos algunas técnicas que mejoran estos resultados, y las compararemos con los resultados anteriores. Finalmente, acabamos el trabajo con las conclusiones y líneas futuras de nuestra investigación.

II. Trabajo relacionado

Las búsquedas epidémicas se estudian en la actualidad para ofrecer recomendaciones de documentos en redes P2P. Además, la definición y clasificación de documentos en un espacio social es un campo abierto de investigación.

II-A. Búsquedas epidémicas

Los algoritmos epidémicos están siendo estudiados con profusión en el campo de la búsqueda de información en redes P2P. El trabajo [5] es una buena introducción a los aspectos más importantes de los algoritmos epidémicos. En este tipo de algoritmos, la información que se quiere intercambiar con la red no se distribuye a todos los nodos que están en contacto con el nodo original, sino solo a un subconjunto escogido mediante algún algoritmo determinado. Los algoritmos epidémicos tienen un comportamiento fuertemente bimodal: a partir de un umbral empinado de los parámetros de la red se asegura la propagación de la información con gran probabilidad.

BuddyCast [1] utiliza un algoritmo simple para hacer búsquedas basadas en paseos aleatorios a través de amigos y amigos-de-amigos. El problema de este algoritmo es que no tiene en cuenta los enlaces que tienen los amigos entre sí, y además la creación de los primeros enlaces a amigos es totalmente manual. Aún así, según los propios autores es un sistema en funcionamiento que permite cientos de miles de usuarios a un tiempo, mejorando indudablemente los sistemas alternativos de inundación. También se propone un parámetro de comparación, la fracción de superposición f_o, que usaremos en el resto de este documento para evaluar el rendimiento de los algoritmos y modelos propuestos. Siendo P el conjunto de peticiones totales de la red, $R_{Epidémico}(p)$ es el conjunto de respuestas de una petición $p \in P$ que ha obtenido un usuario utilizando un algoritmo epidémico y $R_{Exacto}(p)$ el número exacto de recursos en la red que coinciden con la petición p. El parámetro f_o se define como:

$$f_o = \frac{\sum_{p \in P} |R_{Epidémico}(p) \cap R_{Exacto}(p)|}{\sum_{p \in P} R_{Exacto}(p)} \qquad (1)$$

En este trabajo [1] también se presenta una evaluación de un sistema de generación de grupos o clusters de usuarios para mejorar los resultados obtenidos. Pero estos clusters se tienen que crear independientemente de la red y su número está predeterminado. Además, el cálculo de similitud de intereses entre usuarios necesita que los dos usuarios que se están comparando hayan valorado previamente varios recursos comunes. Si el conjunto de recursos que comparten es pequeño la afinidad calculada entre ambos usuarios puede no tener sentido. Si no tienen ningún recurso en común, ambos usuarios simplemente no pueden compararse. La entrada en la red puede ser con este mecanismo muy lenta y laboriosa, ya que un usuario debe valorar cientos de recursos diferentes para maximizar sus probabilidades de encontrar usuarios con quien compararse. Además, como presumiblemente solo se relacionará con los usuarios que hayan valorado sus mismos recursos, se crean grupos cerrados de usuarios que no se relacionan entre sí.

SENSE [3] es un sistema de recomendación en una red centralizada con una base de datos global precompilada. Clasifica los documentos en tres tipos distintos: *sociales*, que son aquellos en el grupo de amigos explícitamente marcado por el usuario; *espirituales*, documentos recomendados por nodos con intereses comunes con el usuario; y *globales*, en donde todos los nodos tienen la misma importancia. La búsqueda y medida de afinidad entre usuarios se realiza a través de superposición de conjuntos comunes de palabras clave. Aunque el sistema de recomendación en el caso de SENSE es centralizado, aprovecharemos la idea de mantener tres grupos diferentes en nuestro trabajo.

II-B. Vectores como perfiles

Una de las representaciones de un documento en el campo de la obtención de información (Information Retrieval, IR) que está teniendo más éxito es la utilización de vectores [8]. Las componentes de cada uno de estos vectores son una frecuencia de aparición de un término determinado en el documento bajo análisis. Una de las métricas más usadas para comparar documentos en este caso es la métrica del coseno, que describiremos más adelante. En [18] encontramos simulaciones de esta técnica de IR con la métrica del coseno, y se demuestra que gracias a ella podemos mejorar los resultados de las búsquedas en P2P.

Los trabajos mencionados solo consideran términos como componentes de estos vectores, pero el estudio se puede generalizar fácilmente a componentes como ontologías o categorías de términos, como se demuestra en [13]. Este estudio ofrece un mecanismo para traducir un vector de términos (*bag-of-words*) a un vector de ontologías o categorías (*bag-of-concepts*) Hacer que las componentes de los vectores pasen a ser términos abstractos soluciona los problemas de sinonimia y polisemia que tienen las palabras de lenguajes humanos.

Una vez que hemos definido los documentos o recursos como vectores, podemos utilizar el mismo espacio vectorial para definir las peticiones de búsqueda. Incluso podemos definir en el mismo espacio vectorial a los propios usuarios, asignándoles el perfil medio entre los recursos que poseen. De esta manera podemos aislar la descripción de los usuarios de la valoración que hacen de sus propios recursos, y al contrario que en propuestas anteriores todos los usuarios de la red P2P pueden compararse entre sí. Aprovecharemos esta ventaja para aumentar las comparaciones entre usuarios, y presentar técnicas adicionales de descubrimiento que no se pueden utilizar en las búsquedas epidémicas clásicas.

III. Modelos, Formalización y Definiciones

En esta sección definiremos los modelos y el escenario de aplicación de nuestro problema, y las asunciones que hacemos en nuestro trabajo.

III-A. Modelo social

En una red P2P hay un conjunto de documentos únicos $R = \{r_1, r_2, \ldots, r_m\}$. Estos documentos o recursos pueden describirse con metadatos, campos e inspección interna de forma que es posible definir conjuntos de palabras, *bag-of-words*, para cada uno de ellos, según [8]. De acuerdo con [13], es posible definir una ontología de clasificación de los recursos en R. Esto es, se obtienen n categorías semánticas con las que podemos describir los recursos en R. Así, cada recurso $r_i \in R$ tiene asociado un perfil dentro de esa ontología $\bar{p}(r_i) = \{c_1, c_2, \ldots, c_n\}$. En este trabajo supondremos que los componentes c_i de $\bar{p}(r)$ son números reales entre 0 y 1. Llamamos **espacio social** $\mathbb{P} = [0,1]^n$ al espacio vectorial de todos los perfiles posibles.

Es posible definir una función distancia $d : \mathbb{P} \times \mathbb{P} \to \mathbb{R}$. En este trabajo utilizaremos la función distancia del coseno [12], que ya se ha usado con éxito para comparar vectores de documentos [8] ya que evita considerar la longitud del documento en el cálculo de la distancia:

$$d(\bar{a}, \bar{b}) = \frac{\bar{a} \cdot \bar{b}}{|\bar{a}||\bar{b}|} = \frac{\sum a_i b_i}{\sqrt{\sum a_i^2 \sum b_i^2}} \quad (2)$$

Un usuario u de la red "posee" o "comparte" un subconjunto de recursos $R_u \subset R$. Supondremos que este usuario tiene para cada recurso $r \in R_u$ un par $(\bar{p}(r), URL(r))$ con el vector descripción del recurso y la URL donde buscarlo. Hay N usuarios, y en nuestro modelo permitimos que dos usuarios u y v puedan tener recursos comunes, $R_u \cap R_v \neq \emptyset$. Podemos asignar a un usuario un perfil $\bar{p}(u) = \frac{\sum_{r \in R_u} \bar{p}(r)}{|R_u|} \in \mathbb{P}$, que es la media de los perfiles de los recursos que el usuario comparte. Así, cada uno de los componente del vector p_u es un número real entre 0 y 1 e indica el grado de interés que un usuario muestra en cada una de las categorías del sistema.

Las búsquedas de los usuarios dentro de la red se hacen por medio de peticiones $\bar{q}$. Como estas peticiones deben poder compararse con las descripciones de los documentos, también pertenecen al espacio social $\bar{q} \in \mathbb{P}$. Suponemos que un usuario u buscará normalmente recursos que están cerca de sus perfiles $\bar{p}(u)$ en el espacio métrico.

Así, tanto los recursos como los usuarios y las búsquedas pueden definirse como vectores en $\mathbb{P}$, y existe una función distancia d para compararlos.

III-B. Modelo de red

Por otro lado, los usuarios se organizan como una red P2P no estructurada. Un usuario u controla un nodo v_u de la red. No es necesario que todos los usuarios estén siempre conectados. Podemos considerar esta red como un grafo simétrico $G = \{V, L\}$, donde V es el conjunto de todos los nodos de la red y L un conjunto de pares con

enlaces entre ellos. Se llama vecindario de un nodo v, Γ_v^1, al conjunto de nodos que tienen un enlace directo con el mismo. Por inducción, se llama Γ_v^i al conjunto de todos los nodos que tienen un enlace a un nodo en Γ_v^{i-1} y no estaban en ningún vecindario inferior. Se llama distribución Λ_v^i al conjunto de nodos unión de todos los vecindarios inferiores a i, $\Lambda_v^i = \cup_{j \leq i} \Gamma_v^j$. Se llama grado de un nodo al número de enlaces que tiene, esto es, $|\Gamma_v^1|$. En nuestro escenario supondremos que el grado de todos los nodos es igual a k, y es mucho menor que el número de nodos en la red N. Llamamos diámetro de la red D a la mayor distancia en saltos de red que separa a dos nodos de la misma. Finalmente, se define el coeficiente de clustering relativo de un nodo v_1 con respecto a otro v_2 como:

$$\gamma_{v_1}(v_2) = \gamma_{v_2}(v_1) = \frac{|\Gamma_{v_1}^1 \bigcap \Gamma_{v_2}^1|}{k} \quad (3)$$

Las redes sociales suelen tener un comportamiento de mundo pequeño [10], [7]. Estas redes tienen un diámetro pequeño mientras que exhiben un coeficiente de clustering grande. Las investigaciones en redes con comportamiento de mundo pequeño mostraron un fenómeno interesante que bautizaron como "la fuerza de los enlaces débiles", y es que el camino entre dos nodos cualquiera de la red pasará con mucha probabilidad por nodos "atajo", que presentan un coeficiente de clustering muy bajo con respecto a sus vecinos. Si suponemos que dos nodos con un alto coeficiente γ forman parte de una comunidad, estos nodos atajos son nodos que no pertenecen realmente a ninguna de las comunidades de la red.

Estos nodos con bajo γ son un atajo entre dos grupos de nodos que no se conocen directamente entre sí, pero que pueden compartir intereses. Llamamos **constelaciones** a cada uno de estos grupos separados en la red pero con intereses comunes. La Fig. 1 muestra un ejemplo de una red con clusters. El usuario está en el centro de su propia visión de la red, y los nodos se clasifican de acuerdo a su distancia en saltos de red. Los nodos similares deberían estar en los anillos centrales para que el proceso de búsqueda resulte más rápido y sencillo. La parte derecha de la figura muestra la distribución real de los recursos de interés dentro de la red, suponiendo que cada usuario tenga varios recursos. En este ejemplo hay una "constelación" de recursos a tres saltos de red del nodo original.

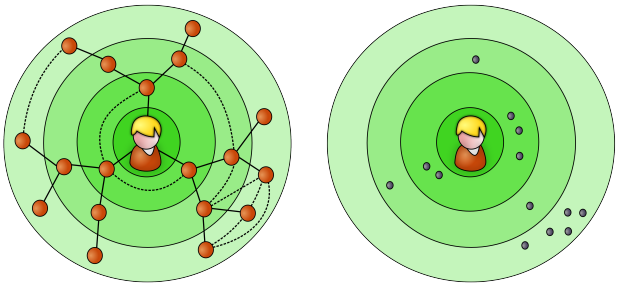


Figura 1. Los vecinos de un nodo y algunas constelaciones de la red

IV. Escenario y Objetivos

Estudiaremos un escenario de distribución y búsqueda de contenido en redes P2P dividido en cuatro fases diferenciadas.

La primera fase es la **creación de la estructura de conexiones** en la red P2P. El objetivo es inducir una red con las características del mundo pequeño, donde los nodos se conectan entre sí atendiendo a la similitud entre sus descripciones. En [17], [2] se demuestra que unas pocas conexiones aleatorias en una red fuertemente agrupada pueden inducir este comportamiento de mundo pequeño.

La segunda fase es la **diseminación de la información** sobre los recursos que un usuario comparte dentro de la red. En nuestro escenario se disemina el par $(\bar{p}(r), URL(r))$ asociado a un recurso $r \in R$. De esta manera, los recursos se pueden encontrar y descargar incluso aunque el nodo que los publicó originalmente en la red desaparezca de la misma.

La tercera fase es la **búsqueda de recursos** en la red basada en afinidades. Un nodo muestra interés por la descripción de un recurso, y busca dentro de la red aquellos que coinciden con esa descripción. Al contrario que los sistemas presentados en las secciones precedentes, en nuestro escenario no imponemos la necesidad de que un usuario evalúe personalmente todos los recursos de la red en los que está interesado, sino que crearemos los grupos de usuarios basándonos en la descripción de los recursos que ya comparten y en la descripción personal de los propios usuarios.

La cuarta y última fase es la **publicación y descarga de los recursos** en sí. El resultado de la fase anterior de búsqueda es una URL desde la que descargar el recurso, y en esta fase se aprovechará la estructura de la red P2P para realizar la descarga.

IV-A. Objetivo

El objetivo de este trabajo es que dada una petición $\bar{q}$ por un usuario u y un umbral $\lambda \in \mathbb{R}$, sea posible encontrar de forma eficiente los recursos de la red que están a una distancia inferior a λ de $\bar{q}$. Esto es, acercarse lo más posible a obtener el siguiente subconjunto máximo:

$$R_q = \{r \in R | d(r, q) < \lambda\} \quad (4)$$

Nos centraremos así en la resolución de la fase primera y tercera de este escenario mediante la creación de una estructura adecuada para las búsquedas. En cuanto a la cuarta fase, la de descarga final del recurso, supondremos que existe un sistema paralelo para que, dada la $URL(r)$ asociada a un recurso r, sea posible obtener el documento físico. Además de permitirnos ignorar el problema de la distribución de recursos, existen motivos de seguridad y eficiencia que lo aconsejan así. En [15], [14] proponemos un sistema de distribución de recursos en redes P2P a partir de su URL.

V. Algoritmo epidémico básico

Las redes P2P que tienen características semejantes a las redes sociales pueden aprovechar el comportamiento de mundo pequeño para mejorar los resultados del mecanismo de búsqueda y ofrecer recomendaciones a los usuarios. En estas redes, los nodos en Γ_u^1 están relacionados con los intereses de un usuario u, además de enlaces aleatorios para mantener la estructura de mundo pequeño. Así, es muy probable que los nodos en Γ_u^1 compartan los mismos intereses que u.

En los algoritmos epidémicos un nodo u no envía la búsqueda a todos los nodos en Γ_u^1, sino que escoge un subconjunto de nodos. Hay tres posibles criterios para escoger este

subconjunto de nodos que recibirán la petición: distancia en el espacio social, coeficiente de clustering y nodos aleatorios, a través de los parámetros M_1, M_2 y M_3 respectivamente.

Los siguientes pseudocódigos muestran el algoritmo de selección de los nodos en el vecindario utilizando los tres parámetros expuestos.

```
foreach r ∈ R_u do
    if d(q̄, p̄(r)) ≤ λ then
        origin.found_resource(URL(r));
    end
end
if hops ≤ H then
    foreach n ∈ select_nexts(q̄, Γ¹) do
        n.search_resource(q̄, origin, hops+1);
    end
end
```

Procedure `search_resource` ($\bar{q}$, *origin, hops*)

```
nexts ← empty_set();
s ← order_by_distance(Γ¹, q̄);
nexts.append(first M_1 elements of s);
s ← order_by_clustering(Γ¹);
nexts.append(last M_2 elements f s);
nexts.append(M_3 random elements of Γ¹));
return nexts
```

Procedure `select_next` ($\bar{q}$)

Los tres criterios de enrutamiento son importantes para el algoritmo epidémico. En primer lugar, el criterio de distancia a través del parámetro M_1 permite encontrar los recursos en los vecinos más cercanos en el espacio social. Es en estos nodos cercanos donde será más fácil encontrar recursos similares, y éste es el mecanismo de selección utilizado en [1], [11]. Además, los nodos cercanos en el espacio social también estarán enlazados entre sí con gran probabilidad. El segundo criterio, a través del parámetro M_2, permite enviar la petición a los nodos que **no** pertenecen al cluster actual del usuario, independientemente de su distancia. Este criterio ha sido poco explorado en el enrutamiento de nodos, aunque es fundamental en la creación de los grupos de [1] donde después se enruta utilizando el primer criterio. En este trabajo evaluaremos su utilización dinámica para enrutar mensajes y no solo en la creación de la estructura. Mediante el criterio de clustering se puede llegar a nodos lejanos que comparten intereses pero no enlaces con los nodos vecinos. Los nodos con bajo coeficiente de clustering son, así, atajos a otras constelaciones de la red. Finalmente, el criterio de nodos al azar a través del parámetro M_3 se utiliza en los protocolos de paseos aleatorios y epidémicos, y también lo evaluaremos en este trabajo.

Las redes sociales reales no empiezan enlazando con "amigos" aleatorios. En realidad, los primeros enlaces de una red P2P social suelen ser los mismos que los de una red social real, entre amigos que ya comparten intereses previamente. Este escenario inicial aumenta espectacularmente el rendimiento de las búsquedas. En cambio, en este trabajo supondremos el caso peor y que no existe un substrato inicial en la red. Por tanto, los primeros enlaces entre nodos serán totalmente aleatorios.

V-A. *Simulación y discusión*

En esta sección simulamos el modelo epidémico descrito en la sección anterior con varios miles de nodos para redes que usen los perfiles de los usuarios como criterio de enlace. Compareremos el rendimiento de este protocolo con el rendimiento de los algoritmos de inundación, que se han utilizado en el pasado para encontrar recursos en redes P2P, como [6], [16]. Aunque no son escalables con el número de nodos y las búsquedas pueden llegar a consumir gran cantidad del ancho de banda de la red, por sus características pensamos que la comparación de los algoritmos epidémicos con el ideal de algoritmo de inundación es interesante para comprobar su rendimiento real. Para realizar las simulaciones de esta sección utilizamos una modificación de Peersim [4] que es capaz de llegar a simular redes de decenas de miles de nodos.

Definimos **rendimiento** del algoritmo de búsqueda como la media de la fracción de superposición f_o de la Ec. 1 para todos las búsquedas realizadas dentro de la red, es decir, el cociente entre los recursos encontrados y los realmente existentes en la red, sin tener en cuenta los falsos negativos. Este parámetro no incluye las búsquedas sin resultado, ya que será otro parámetro a analizar más adelante.

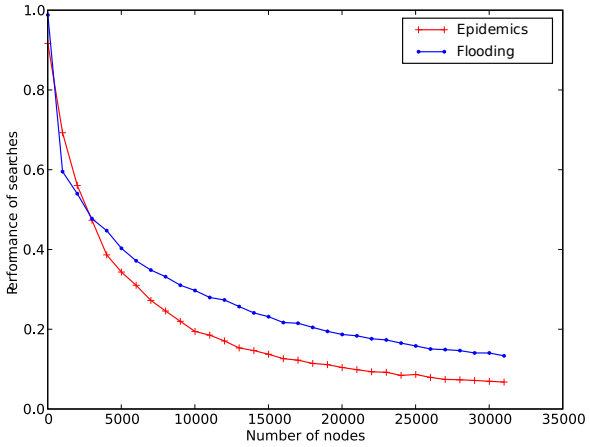
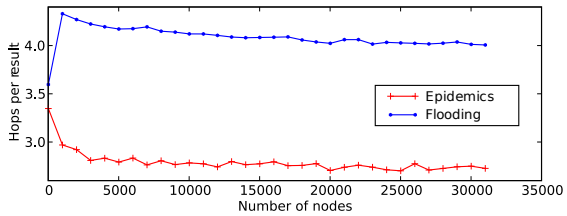
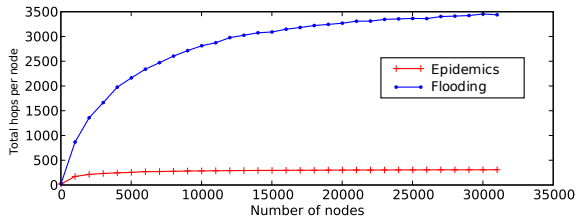


Figura 2. Rendimiento variando el número de nodos en la red

La Fig. 2 muestra el rendimiento de los algoritmos analizados con un número creciente de nodos en la red y una búsqueda única para cada uno de ellos. Ya que el diámetro de la red se incrementa con el número de nodos en la misma, y el número máximo de saltos H está limitado, los dos algoritmos analizados tienen un rendimiento decreciente con el número de nodos. Eso no significa que un usuario obtenga menos resultados en sus búsquedas, sino que hay más recursos en la red que se corresponden con la descripción buscada. Supondremos que un usuario queda satisfecho al encontrar algunos recursos en la red, y no necesita encontrarlos todos. En la parte derecha de la Fig. 2, que se corresponde con una red de más de 30.000 nodos, el algoritmo de inundación limitado a $H = 5$ saltos puede localizar el 20 % de los recursos de interés en la red, mientras que el algoritmo epidémico solo encuentra el 10 % de estos recursos. A pesar de su mejor rendimiento, como veremos más adelante el coste en mensajes de red del algoritmo de inundación es



(a) Saltos medios por búsqueda



(b) Saltos totales (incluyendo todas las búsquedas con y sin resultados)

Figura 3. Uso de la red del algoritmo epidémico básico

inaceptable.

Es interesante comparar los resultados de la Fig. 2 con la Fig. 3(a). Esta nueva figura muestra el número medio de saltos necesarios para encontrar un recurso a medida que aumentamos el número de nodos en la red, para $H = 5$. Como esperábamos, el algoritmo de inundación necesita muchos más saltos que los algoritmos epidémicos para encontrar un recurso. Además, el número medio de saltos para los algoritmos epidémicos está cerca de 1. Esto significa que los usuarios solo encuentran los recursos de interés dentro de su propio cluster, definido en este caso como Γ^2. Una ventaja del pequeño número de saltos necesario con el modelo epidémico es que el descubrimiento es realmente rápido y no se necesita sobrecargar la red con mensajes.

La Fig. 3(b) también muestra el uso que hace la red de ambos algoritmos, definido como el número de saltos necesarios para realizar una búsqueda incluso cuando ésta no ha tenido éxito. Como esperábamos, el algoritmo de inundación necesita muchos más mensajes que los algoritmos epidémicos. Así, la relación de rendimiento de las búsquedas contra uso de la red se incrementa con el número de nodos, mientras que los algoritmos epidémicos son escalabes con el número de nodos.

Los algoritmos epidémicos básicos se comportan mucho mejor que los algoritmos de inundación en cuanto a uso de la red y velocidad de las búsquedas, pero tienen un rendimiento bajo comparado con la inundación. Intuimos que este rendimiento tan bajo se debe a la existencia de constelaciones no detectadas en la red, y creemos que es posible mejorar estos resultados mediante algunas modificaciones en el modelo epidémico que presentaremos en la siguiente sección.

VI. Algoritmo epidémico mejorado

En esta sección proponemos mecanismos adicionales sobre el modelo epidémico orientados a mejorar el rendimiento de las búsquedas sin perder la ventaja de su bajo uso de la red.

VI-A. *Fuzzy Distributed Hash Table*

Los métodos explicados en la sección V pueden crear grupos aislados de nodos que comparten intereses, creando las constelaciones de la Fig. 1. En estos casos, los algoritmos de búsqueda epidémica pueden no ser suficientes para encontrar estas constelaciones aisladas, y las conexiones aleatorias como atajos no siempre funcionan. Para prevenir la creación de estas constelaciones, proponemos la creación de tablas de hash distribuidas borrosas (Fuzzy Distributed Hash Table, FDHT).

En las tablas de hash distribuidas tradicionales (DHT), los clientes usan una clave determinada para acceder a un único valor. Todos los accesos con la misma clave acabarán devolviendo la misma información. La estructura de la DHT se distribuye entre los nodos de la red, y cada uno de ellos está a cargo de guardar un rango de claves. Kademlia [9] es un ejemplo bien conocidos de DHT.

En una FDHT como la de nuestra propuesta, el espacio de claves es mucho más pequeño que el espacio de nodos que las guardan. Así, varios nodos se encargan de guardar los datos asociados a la misma clave, y una petición de datos bajo esa clave puede devolver la información almacenada en cualquiera de estos nodos.

Supongamos una DHT con N nodos que almacena $e < N$ claves. Aunque las DHT tradicionales están diseñadas para $e \gg N$, se pueden usar en este caso, con la ventaja adicional de que se pueden relajar las restricciones de replicación de información y enrutamiento. Así, cada clave k del espacio de claves de la DHT estará asignada un conjunto de nodos C_k tal que de media $E[|C_k|] = N/e > 1$. Con los algoritmos relajados de la DHT, una inserción bajo la clave k puede acabar en cualquier nodo del conjunto C_k, mientras que una lectura de k puede responderse desde cualquier nodo en C_k. El resultado es que pueden almacenarse varios valores bajo la misma clave k, y cada consulta puede responderse con cualquiera de esos valores. Este esquema de funcionamiento es una FDHT.

Como ejemplo de implementación, describiremos cómo se puede modificar una DHT tipo Kademlia [9] para cubrir los objetivos de una FDHT. En Kademlia cada nodo guarda k cachés (llamada cada una de ellas *k-bucket*) de n nodos diferentes. Cada k-bucket guarda solo direcciones a nodos a una distancia determinada del identificador de red del nodo. Cuando desea hacer una búsqueda de una clave, pregunta a m nodos del k-bucket correspondiente a la petición. Si alguno de ellos conoce la respuesta, la devuelve inmediatamente. Si no la conoce, devuelve la dirección de m nodos que tienen más posibilidades de conocer la respuesta al estar más cerca del identificador de la petición. Este algoritmo funciona suponiendo que m es un número relativamente grande de nodos, y que solo los nodos más cercanos a la clave de la petición guardarán el valor asociado. La forma de rellenar los k-buckets se basa en listas FIFO, con un número n relativamente grande. Sobre esta red podemos conseguir un comportamiento FDHT si usamos un número n pequeño que obligue a una rotación muy rápida de las direcciones de los nodos en los k-buckets, un número m pequeño para que el nodo consulte a un conjunto pequeño de nodos remotos cada vez, y a relajar los criterios de cercanía a una clave para que un nodo pueda almacenar el valor. De esta manera, peticiones

y publicaciones con la misma clave pueden acabar en nodos diferentes, y por tanto se puede asignar un conjunto de valores a cada una de las claves. Es este escenario, el valor devuelto por una petición k puede ser cualquiera de los elementos del conjunto C_k asociado.

Veamos a continuación cómo utilizamos la FDHT para mejorar las búsquedas en nuestro escenario. Supongamos que todos los nodos en la red comparten un conjunto de perfiles $E = \{p1, p2, ..., p_e | p_i \in \mathbb{P}\}$, donde $e = |E|$ es una potencia de 2. Estos perfiles están equiespaciados en el espacio social, y los llamamos "perfiles ejemplo". Todos los nodos de la red comparten este conjunto E, y pueden calcular cuál es el perfil ejemplo más cercano a su propia descripción. Supongamos también que existe una función $e(p \in E)$ que es capaz de transformar un perfil ejemplo en una clave uniformemente distribuida en la FDHT. La implementación de esta función puede ser simplemente un hash del perfil ejemplo.

En la arquitectura propuesta, un nodo utiliza la FDHT para descubrir nuevos nodos que están cerca de su perfil. Periódicamente, un nodo inserta su propia dirección bajo la clave del perfil ejemplo más cercano a su propia descripción, y realiza la petición de la misma clave. Aprovechando que las respuestas de la misma clave no son únicas, la respuesta a la petición del perfil ejemplo más cercano puede ser un vecino que el nodo ya conoce, él mismo o un nuevo nodo completamente nuevo también cercano al mismo perfil de ejemplo. Probaremos en las simulaciones que la inclusión de una FDHT ayuda a mejorar el rendimiento del algoritmo de búsqueda.

VI-B. Criterios para aceptar nuevos nodos

En la arquitectura propuesta hay un límite superior para el número de enlaces que puede tener un nodo. Los nodos pueden desaparecer de la red porque se van, dejan de funcionar o hay fallos de comunicación. Los protocolos P2P tienen mensajes PING para detectar estas desapariciones, y los nodos tienen así espacios libres en su lista de enlaces para llenarlos con nuevos nodos. Aún así, la velocidad de descubrimiento de nuevos nodos es usualmente mayor que la velocidad de desaparición de los antiguos, así que muchas veces los nodos desearán enlazar con un amigo nuevo pero no tendrán disponible ningún enlace.

Las soluciones clásicas para borrar enlaces, como las colas FIFO y LIFO, no funcionan bien en redes sociales. Como se ha comprobado en las secciones anteriores, la característica más importante de nuestra red social es que hay enlaces a nodos muy distintos que sirven como atajos a otras constelaciones. Además, la creación de una red de mundo pequeño necesita de estos atajos. En este sentido, creemos que los enlaces deben eliminarse de acuerdo con las características medias de la red, nunca de los extremos, y no de acuerdo con el uso real que se hace de los mismos, siempre con el objetivo de mantener una red con las características de mundo pequeño.

Hay dos características en la red social que un nodo u puede utilizar para eliminar su enlace con otro nodo v, la distancia social entre ambos $d(u, v)$ y el coeficiente de cluster relativo $\gamma_u(v)$. Si el coeficiente $\gamma_u(v)$ es alto, quiere decir que hay otros nodos en Γ_u^1 que enlazan a v. De esta manera, perder el enlace con v no significa perder demasiada información, pues aún podríamos contactarlo a través de algún otro nodo del vecindario si $H > 2$. Entonces, u puede borrar con seguridad los nodos con alto $\gamma_u(v)$. De una forma similar, los nodos que están cercanos a u con mucha probabilidad estarán también cercanos a algún nodo de Γ_u^1, y con mucha probabilidad serán accesibles también a través de los nodos del vecindario. Aún así, los nodos con una distancia social pequeña a u y que además tengan un coeficiente $\gamma_u(v)$ pequeño, son posiblemente enlaces a otras constelaciones de la red con las que no tenemos comunicación directa, y si u borrase estos nodos podría reducir la eficiencia de las búsquedas.

El mecanismo propuesto tiene en cuenta estos dos criterios, distancia y clusters, para borrar los enlaces de la lista. Cuando un nodo recibe una petición de enlace desde otro nodo y tiene su propia lista de vecinos completa, calcula la distancia d y el coeficiente de cluster c al nuevo nodo y compara estos valores con la **mediana** de los valores que ya conoce en Γ_u^1. Si d es menor que la mediana de distancias y c es menor que la mediana de coeficientes de cluster, el nuevo nodo es aceptado en la lista, tal como muestra el siguiente pseudocódigo.

```
d ← get_distance(node);
c ← get_clustering(node);
median ← len(neighbors)/2;
if d < order_by_distance(neighbors)[median] & c <
order_by_clustering(neighbors)[median] then
    remove_further_node(neighbors);
    add_link(node, neighbors);
    return True
end
return False
```

Procedure `link_to_node(`*node*`)`

VI-C. Simulación

La tabla I muestra la distribución de recursos en el vecindario de un nodo de la propuesta comparado con una red aleatoria. En esta simulación usamos una red de $N = 1000$ nodos, cada uno conectado a $k = 5$ vecinos, y esperamos hasta que cada nodo ha realizado al menos 4 búsquedas. Al final de la simulación escogemos un nodo al azar u que suponemos representativo y calculamos la longitud de su vecindario, distribución de secuencia y el porcentaje de recursos de su interés que están a cada salto de red. Repetimos esta simulación con una red creada enteramente al azar.

Como muestra la tabla, el algoritmo de creación de estructuras es capaz de mantener el 94 % de los recursos de interés de un usuario a menos de 3 saltos de red, y en este caso solo necesita contactar con 41 nodos para encontrarlos. La red aleatoria necesita, como comparación, 6 saltos de red para alcanzar un porcentaje similar, y contactar con prácticamente todos los nodos de la red. Por otro lado, la grupos en la red han aumentado el diámetro de la misma hasta 12, mientras que la red al azar tiene un diámetro más limitado de 7. El algoritmo de clusters aumenta el diámetro de la red, e incluso crea islas de nodos aislados que no pueden ser contactados. Es muy posible que estos nodos no tengan recursos de interés para el resto de la red.

La Fig. 4 es equivalente a la Fig. 2, pero ahora se incluye el rendimiento de las propuestas de mejora de algoritmos

Cuadro I
NETWORK STRUCTURE. $N = 1000$ $k = 5$ $M_0 = M_1 = M_2 = 1$ $p = 0{,}5$

	Cluster			Random		
i	$\|\Gamma_v^i\|$	$\|\Lambda_v^i\|$	$\%res$	$\|\Gamma_v^i\|$	$\|\Lambda_v^i\|$	$\%res$
1	6	7	18.18	6	7	0
2	12	19	58.82	20	26	7.14
3	22	41	94.11	78	104	21.42
4	48	89	100	245	349	42.58
5	100	189	100	46.3	812	85.71
6	156	345	100	186	998	100
7	224	569	100	2	1000	100
8	226	795	100	-	-	-
9	156	951	100	-	-	-
10	33	984	100	-	-	-
11	8	992	100	-	-	-
12	1	993	100	-	-	-
13	0	993	100	-	-	-

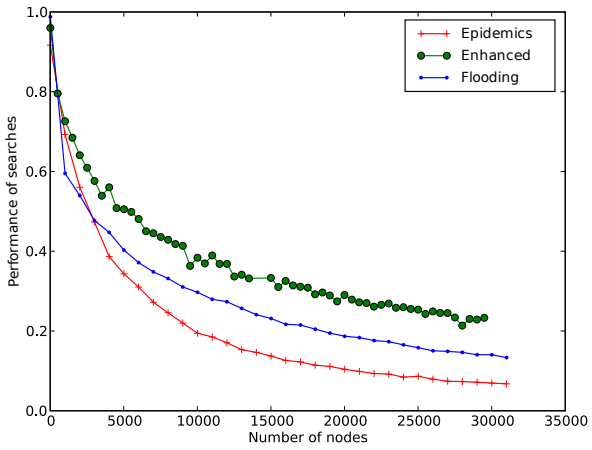
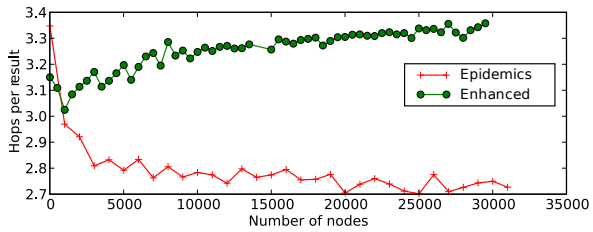


Figura 4. Rendimiento contra el número de nodos en la red

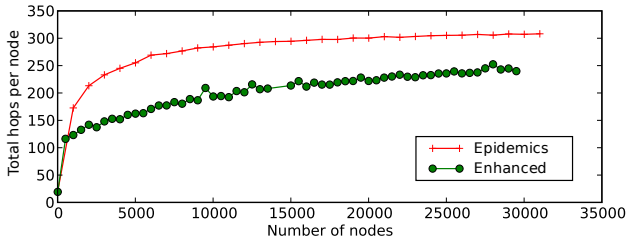
epidémicos con una FDHT con 10 perfiles de ejemplo. Se muestra que las mejoras no solo incrementan el rendimiento de los algoritmos epidémicos básicos, sino que incluso mejora el rendimiento del algoritmo de inundación. Esto es posible gracias a la estructura de la red creada por la FDHT, ya que es capaz de descubrir constelaciones y mantener así los recursos de interés más cerca de los usuarios.

La Fig. 5(a) es la misma que la Fig. 3(a), incluyendo los resultados de las modificaciones propuestas para una FDHT con 10 perfiles de ejemplo. Como muestra la figura, las modificaciones propuestas aumentan el número de saltos que un mensaje de búsqueda necesita para tener éxito. Esto es debido a que las modificaciones alientan la creación de grupos más amplios, por encima del número máximo de vecinos de un nodo k. A cambio, los algoritmos epidémicos básicos necesitan enviar más mensajes a la red, ya que conocen menos nodos afines al que realiza la búsqueda. Estas figuras muestran que las modificaciones propuestas aún son escalables con el número de nodos, e incluso mejoran el uso de la red de los algoritmos básicos.

La Fig. 6 muestra, para una búsqueda simple en todos los nodos, el porcentaje de búsquedas sin resultado para un número creciente de nodos en la red. Para probar si los algoritmos son capaces o no de encontrar recursos poco



(a) Saltos medios por búsqueda



(b) Saltos totales (incluyendo todas las búsquedas con y sin resultados)

Figura 5. Uso de la red de las propuestas

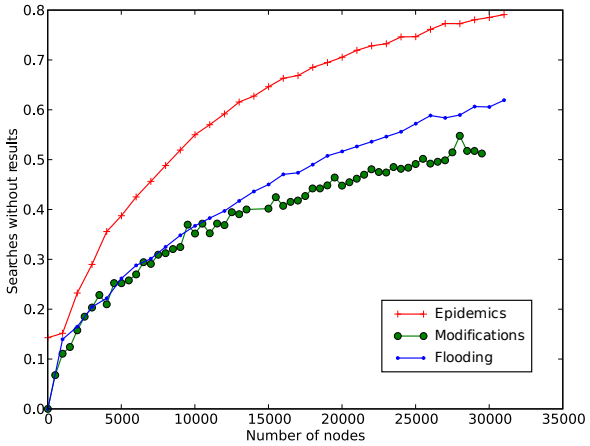


Figura 6. Búsquedas sin resultados

comunes, el número de recursos que coinciden con una búsqueda se ha mantenido intencionadamente bajo. Aún así, el alto número de búsquedas sin resultados para los algoritmos básicos es remarcable, y se debe al efecto de los grupos cerrados y las constelaciones. Muchos de los nodos de la red tienen problemas para encontrar otros nodos que comparten sus mismos intereses. Aquí es precisamente donde la FDHT ayuda, como muestra la figura. De todas formas, el número de recursos encontrados aumentará en todo caso a medida que los usuarios aumenten el número de búsquedas, como veremos a continuación.

Las Fig. 4 y 6 prueban que la rápida convergencia de las modificaciones es válida para un escenario en que haya pocas búsquedas o cuando la entrada y salida de los usuarios de la red es muy rápida.

La Fig. 7 muestra el rendimiento del algoritmo con un número creciente de búsquedas en una red de 5.000 nodos.

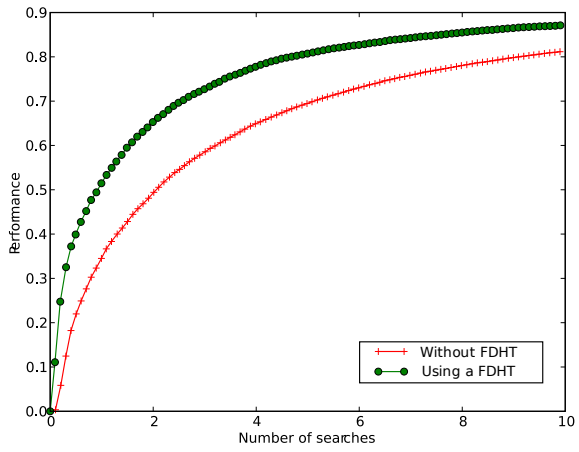


Figura 7. Rendimiento contra el número de búsquedas en cada nodo

Cuando el número de búsquedas se incrementa, el conocimiento de la red también se incrementa, igual que el número de nodos similares encontrados. El rendimiento así se incrementa igualmente, y en este escenario después de solo 10 búsquedas ya es posible encontrar al 80 % de los recursos. Así, ambas versiones de los algoritmos epidémicos se comportan similarmente en el largo plazo.

VII. Conclusiones y trabajo futuro

En este artículo hemos analizado cómo los algoritmos epidémicos se pueden utilizar para encontrar recursos en una red P2P no estructurada y basada en afinidades por perfiles de usuarios, mediante la creación y aprovechamiento de redes con características de mundo pequeño.

A continuación modelamos un algoritmo epidémico básico y demostramos que utiliza una cantidad mucho menor de recursos de red que simplemente la inundación de la red, a costa de un peor rendimiento. Este peor rendimiento puede mejorarse estudiando las características de las redes creadas. Una tabla de hash distribuida difusa permite encontrar rápidamente las constelaciones de recursos de interés que están inicialmente lejos de los usuarios en saltos de red, mientras que la modificación del algoritmo de bajas en la lista de vecinos permite no perder información del cluster en el que se encuentra actualmente el usuario.

En la sección VI-C simulamos estos mecanismos propuestos en redes con miles de usuarios, y demostramos que mejoran la eficiencia del protocolo epidémico básico, permitiendo un tiempo de convergencia hacia el máximo mucho más rápido mientras que tienen un consumo comparable de recursos de red y siguen siendo escalables.

Hay una importante línea futura de investigación en estas redes. Los intereses de los usuarios contienen mucha información que los propios usuarios quieren mantener privada. El lector notará que ninguna de las redes propuestas hasta el momento toma en consideración la privacidad del usuario. Los autores intuyen que el cambio de concepto en las afinidades, comparando perfiles de los usuarios en vez de recursos compartidos, facilita la gestión de la privacidad del usuario, como esperan demostrar en el futuro.

Agradecimientos

Este trabajo ha sido financiado por el proyecto CICYT SECONNET (TSI2005-07293-C02-01) y CONSOLIDER ARES (CSD2007-00004).

Referencias

[1] Amelie Anglade, Marco Tiemann, and Fabio Vignoli. Complex-network theoretic clustering for identifying groups of similar listeners in p2p systems. In *RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems*, pages 41–48, New York, NY, USA, 2007. ACM.

[2] Francesc Comellas and Michael Sampels. Deterministic small-world networks. *Physica A: Statistical Mechanics and its Applications*, 309(1-2):231–235, 2002. Small-world networks;.

[3] Tom Crecelius, Mouna Kacimi, Sebastian Michel, Thomas Neumann, Josiane Xavier Parreira, Ralf Schenkel, and Gerhard Weikum. Making sense: socially enhanced search and exploration. *Proc. VLDB Endow.*, 1(2):1480–1483, 2008.

[4] Andreas Deutsch, Niloy Ganguly, Tore Urnes, Geoffrey Canright, and Márk Jelasity. Implementation for advanced services in ahn, p2p networks. Technical report, Università di Bologna, January 2005. BISON Project, IST-2001-38923.

[5] Patrick Euster, Rachid Guerraoui, Anne-Marie Kermarrec, and Laurent Maussoulie. From epidemics to distributed computing. *IEEE Computer*, 37(5):60–67, May 2004.

[6] M. Jovanovic, F. Annexstein, and K. Berman. Scalability issues in large peer-to-peer networks - a case study of gnutella. Technical report, University of Cincinnati, 2001.

[7] Mei Li, Wang-Chien Lien, and A. Sivasubramaniam. Semantic small world: an overlay network for peer-to-peer search. In *12th IEEE Internation Conference on Network Protocols*, pages 228–238, October 2004.

[8] Christopher D. Manning, Prabhakar Raghadan, and Hinrich Schütze. *An Introduction to Information Retrieval*. Cambridge University Press, April 2009.

[9] P. Maymounkov and David Mazières. Kademlia: a peer-to-peer information system based on the xor metrid. In *IPTPS*, pages 53–65, 2002.

[10] J. A. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H. J. Epema, M. Reinders, M. R. van Steen, and H. J. Sips. Tribler: a social-based peer-to-peer system: Research articles. *Concurr. Comput. : Pract. Exper.*, 20(2):127–138, 2008.

[11] J.A. Pouwelse, J. Yang, M. Meulpolder, D.H.J. Epema, and H.J. Sips. Buddycast: An operational peer-to-peer epidemoc protocol stack. In *Fourteenth Annual Conference of the Advanced School for Computing and Imaging*, June 2008.

[12] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Conference on Computer supported Cooperative Work*, pages 175–186, New York, NY, USA, 1994. ACM.

[13] Duc Thanh Tran, Stephan Bloehdorn, Philipp Cimiano, and Peter Haase. Expressive resource descriptions for ontology-based information retrieval. In *Proceedings of the 1st International Conference on the Theory of Information Retrieval (ICTIR'07), 18th - 20th October 2007, Budapest, Hungary*, pages 55–68, 2007.

[14] Juan Vera-del-Campo, Juan Hernández-Serrano, and Josep Pegueroles. Análisis de seguridad en un sistema de archivos distribuido. *X Reunión Española sobre Criptografía y Seguridad de la Información (RECSI)*, 2008.

[15] Juan Vera-del-Campo, Juan Hernández-Serrano, and Josep Pegueroles. Scfs: Design and implementation of a secure distributed filesystem. *SECRYPT*, 2008.

[16] Juan Vera-del-Campo, Josep Pegueroles, and Miguel Soriano. Msd: A middleware for secure service discovery in pervasive and mobile computing environments. *Journal of Networks*, 2007.

[17] Duncan J. Watts. *Small Worlds: The Dynamics of Networks between Order and Randomness*. Princeton Studies on Complexity, 2003.

[18] Wai Gen Yee, Dongmei Jia, and Ophir Frieder. Finding rare data objects in p2p file-sharing systems. *Proceedings - Fifth IEEE International Conference on Peer-to-Peer Computing, P2P 2005*, 2005:181–190, 2005. Peer-to-peer (P2P) file-sharing systems;Search functionality;Ranking metric;Rare data objects;.

Benefits on using H-P2PSIP in mobile environments

Isaias Martinez-Yelmo*, Alex Bikfalvi†, Carmen Guerrero*

*Departamento de Ingeniería Telemática
Universidad Carlos III de Madrid
Av. Universidad 30
28911 Leganés. Madrid (Spain)
Email: {imyelmo, guerrero}@it.uc3m.es

†IMDEA Networks
Av. del Mar Mediterráneo 22
28918 Leganés. Madrid (Spain)
Email: alex.bikfalvi@imdea.org

***Resumen*—The use of peer-to-peer technologies is increasing everyday and the improvement of mobility technologies is a reality. Now, it is expected that peer-to-peer applications run on mobile devices, but the conjunction of these two technologies is an open research issue. The user mobility impacts on the churn suffered by peer-to-peer networks and consequently it impacts on their performance. Therefore, some mechanisms are necessary to minimize this undesirable effect. Our proposal tries to solve this problem by using a Hierarchical P2PSIP architecture where different overlays are used for different peer mobility behaviours and they are interconnected between them through an interconnection overlay. In this way it is possible for peers that share the same behaviour to choose a certain protocol or to optimize some functionality that suits best with their mobility situation, while maintaining connectivity with all peers.**

***Palabras Clave*—H-P2PSIP, P2PSIP, DHT, Mobility, Performance**

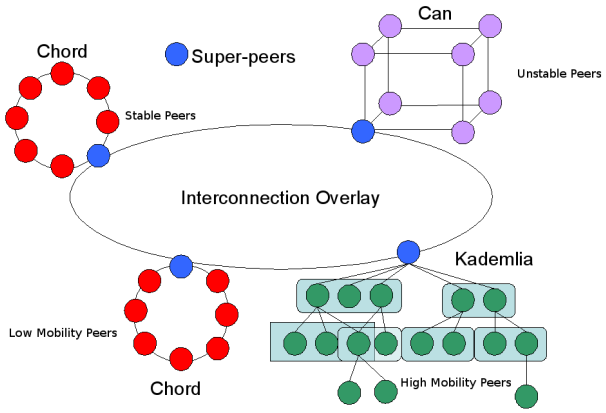


Fig. 1. Hierarchical DHT connecting domains with different peer behaviours

I. Introduction

Peer-to-peer technologies have had a great impact in the Internet in recent years. These peer-to-peer technologies present a scalable solution for distributed services such as file sharing, Voice over IP (VoIP), Video on Demand (VoD), Instant Messaging (IM), etc. Nowadays there are several peer-to-peer applications with great impact; Skype [1], [2] is one of the most successful. However, it is a proprietary solution that is not based on any standard. An open standard like Session Initation Protocol (SIP) would be desirable but in a decentralised fashion instead of the current server based solution.

The IETF P2PSIP[1] Working Group is working on a new protocol to offer an open standard in this field. P2PSIP [3] defines a peer-to-peer overlay-based solution that enables a decentralised architecture which is specially focused, but not only, in replacing SIP. It is expected to standardise a flexible protocol [4] which will be able to support most of the Distributed Hash Tables (DHTs) that can be found in the literature [5], [6], [7], ...

[1]http://www.p2psip.org

However, not only the evolution and deployment of peer-to-peer technologies is increasing everyday, the mobility based on 3G technologies and next 4G networks presents a more complex scenario. It must be taken into account the cross-effects among peer-to-peer overlay networks and mobility. If both technologies are used together, the continuous change in the devices location due to the itinerancy increases the churn and affects to the peer-to-peer performance [8]. Therefore, some optimisations and new proposals are needed for this type of mobile environments which would be a key factor in the near future. Our proposal takes advantage of the H-P2PSIP architecture in order to give a different treatment to peers with different mobility behaviours. In fact different overlays are created where the peers on an overlay have the same peer mobility behaviour (see Figure 1). This separation not only allows tuning the configuration parameters of each overlay network according the behaviour of its peers, it also allows to choose the most suitable overlay according to their behaviour under churn [8]. Therefore this approach opens a new dimension of research in the scenario of peer-to-peer networks deployed under mobile scenarios: which is the most

suitable overlay and the best setup parameters to obtain the best performance in a specific scenario.

The paper is structured as follows. Section II explains how the mobility affects to the peer-to-peer networks. In section III, an overview of the state of the art of P2PSIP is given and in section IV a short overview of mobility in order to put in context our proposal. The different peer mobility behaviours that can be considered in this design and how to manage them are treated in section V. Finally, the proposal of interconnecting different P2PSIP domains with different peer mobility behaviours is explained in section VI. Section VII addresses the conclusions and the future work.

II. Problem Statement and Related Work

One of the main problems in peer-to-peer networks is the stabilisation of peer-to-peer routing tables in order to maintain an average number of hops towards a desired destination. The mechanisms to update these routing tables can be optimised [9] by taking into account the expected churn of the peers. A trade-off exists between complexity or traffic overhead and freshness of the routing tables. However, the evaluation of this trade-off is not a trivial issue. Works like [10], [11] and [12] have collected data from different peer-to-peer networks and have found that although many peers have a significant churn, there is also a set of peers which are really stable. Depending on the level of churn, different strategies can be adopted in order to maintain updated routing tables in the peers [9]. However, not all the peers present the same churn, thus it is difficult to obtain the optimal setup parameters. Other approach is [13] where the peers with high churn don't participate in the maintenance on the overlay because they cause more drawbacks than benefits. These peers can retrieve the information from the overlay as far as their instability let them to do it. With this approach fake routing entries are avoided and a better performance is obtained, although the peers that support the overlay have to increase their work load.

Furthermore, if we take into account the mobile environments and the disruptions caused by the mobility process [14], both the churn of peers and the maintenance overhead of the routing tables in the peer-to-peer networks increase. Therefore, the problem in mobile peer-to-peer networks is how to manage efficiently these peer-to-peer networks where different peer mobility behaviours exist. This efficient management consists in minimising the cost of maintaining the routing tables. This maintenance in mobile environments is not trivial because mobility has a great impact in the network conditions: new IP addresses, new topological points of attachment, different bandwidth conditions or different Round Trip Times (RTT's). Some proposals, like [13] as mentioned above, remove the peers with high churn in the maintenance tasks. However, in this paper we propose an architecture that provides a mechanism which allows dealing with this type of environments more efficiently with a higher flexibility.

Around this topic of peer-to-peer technologies in mobile environments, there is not very much work yet because of its high complexity. There are some works like [15] that establish the requirements needed for peer-to-peer networks in mobile environments. Basically, because of the problems associated with mobility, specially the increment in the churn, it is necessary to provide mechanisms that increase the traffic overhead, the churn itself, etc, All these requirements can be summarised in one: to increase the scalability as much as possible in order to reduce the drawbacks of a mobile scenario. Furthermore there are some solutions that take into account mobile ad-hoc networks like [16]. However this solution is very coupled to the routing infrastructure and to the movement patterns of nodes, which makes to look one of the advantages of overlay networks, applicability under a great variety of scenarios and conditions.

III. P2PSIP

The target of IETF P2PSIP WG is to develop a protocol that can support any DHT overlay network. The aim of this design is to allow an easy deployment of distributed services. The protocol allows to locating resources, services and users in a decentralised way. The first usage that can be used to this protocol is for obtaining a decentralised SIP service, although it can be used for other purposes.

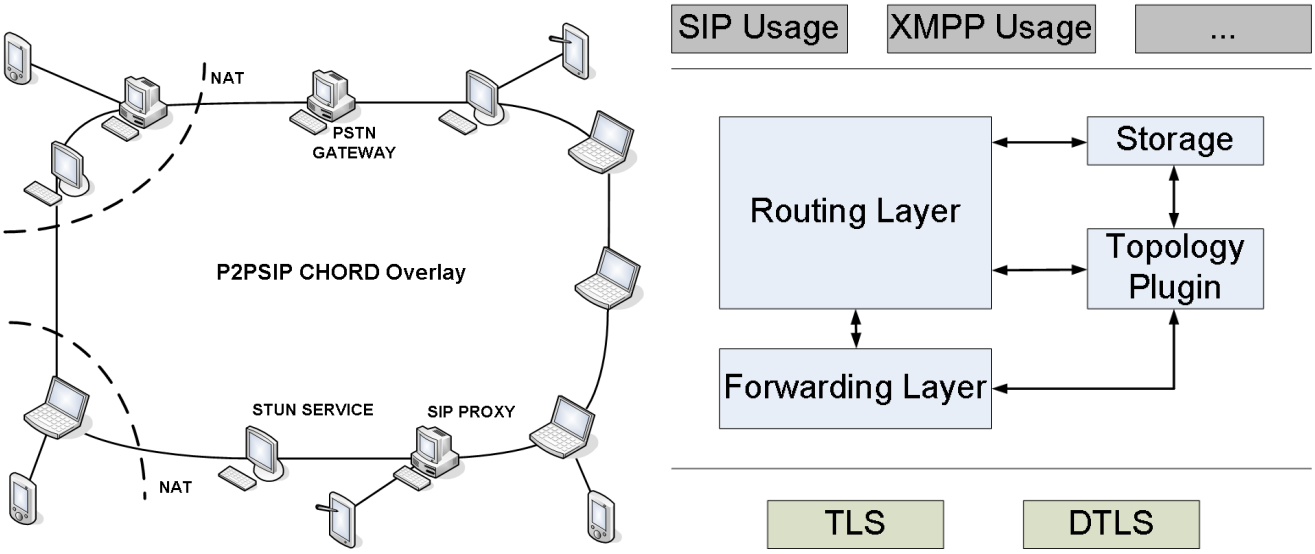
Figure 2(a) presents the P2PSIP Overlay Reference Model using the basic concepts from [3]. P2PSIP protocol is designed to support any type of DHT-based network. Each deployed overlay network is identified by an Overlay ID and the nodes in the overlay can be peers or clients. Peers are active node participants in the overlay network and they are uniquely identified by a Node ID (e.g. the computers and laptops in Fig.2(a)). On the other hand, clients are entities that use the resources offered by the peer-to-peer overlay network but they do not participate in its maintenance. This role should be only used by devices with very limited capabilities, such as the handheld devices shown in Fig. 2(a). The resources in the overlay are uniquely identified by a Resource ID. These resources can be composed by several items like data, files, service references, etc. Peers help to maintain all this information in the overlay and any peer or client in the overlay can retrieve this information. In order to fulfil the requirements, a set of primitives have been defined such as joining, bootstrapping, resource allocation and maintenance. The RELOAD protocol [4] is being defined to implement these tasks with a modular design supporting different overlays and applications (see Fig. 2(b)).

IV. Mobility

Mobility is a characteristic that is being more usual in user terminals and devices. This feature allows connectivity wherever and whenever some access technology is available for it. This feature can be summarised with the famous concept *always-on*. Although to provide mobility is common property nowadays, it is not a trivial functionality. Many studies have been performed to achieve seamless mobility; however this fact is impossible to obtain completely. It is usual to have some disruption in the communication availability of a terminal when is changing from one cell to another or from a technology to another.

A. Macro-mobility and Micro-mobility

A first classification of mobility support solutions that can be considered is the existence of macro-mobility and micro-mobility. Many definitions can be used to explain these terms, but in a simple way we consider them as follows.



(a) P2PSIP Overlay Reference Model (b) P2PSIP protocol reference model

Fig. 2. P2PSIP reference models

Macro-mobility is the mobility of terminals between different domains. The concept of domain here is quite wide. But in this context we mean a part of the network where mobility can be managed with a local solution, a micro-mobility solution.

Micro-mobility happens when the movement is performed inside a domain (i.e. adjacent cells of the same network). Thus, micro-mobility manages mobility closer to the terminal and implies a faster resolution of the connectivity disruption in the terminals. Several proposals have been studied to solve this problem [14], [17], [18]. These types of mobility are usually associated to the access technology used by the terminals. For instance, in UMTS, micro-mobility implies the management of changing from one cell to another, whereas macro-mobility implies the movement of a terminal from one operator network to another or changing from one access technology to another. The time needed for macro-mobility handovers is larger with respect to micro-mobility handovers.

B. Mobile IP

Mobility in IP networks imply the need to change the IP address of the moving terminal each time it moves to a new network. Micromobility solutions can hide or avoid this change of the IP address if the movement is within a micromobility domain. But in other cases, i.e. without a micromobility solution or when changing the micromobility domain, the terminal needs to change the IP address when moving. The reason is that IP addresses act as locators of the terminal, and must have a value according to where the terminal is connected to the network.

An additional problem is that IP addresses are not only locators, they also act as identifiers. This means that to keep ongoing communications, a moving terminal requires a permanent IP address as part of the identifier of its communications. The IETF[2] has standardised solutions to support IP mobility both for IPv4 [19] and IPv6 [20] that work by associating with the terminal a permanent address that acts as identifier (the home address, HoA), and temporal addresses that act as locators and that the terminal configures in the visited networks (Care of Addresses, CoA). A new entity, the Home Agent (HA) is introduced to act as rendezvous point for the communications of the terminal using the HoA. The HA is situated where the HoA is topologically valid and forwards packets to the mobile terminal. Furthermore, in order to accelerate the signalling with the HA, a strategy based on anchor points can be adopted [18]. It must be considered that these optimisations must be done per each flow that it was established before the movement.

[2]http://www.ietf.org

V. Peer-to-Peer Overlays and Mobility

Once that the topic of mobility has been shortly reviewed in the previous section, we can consider how affects to DHT overlays which are supported by P2PSIP.

The mobility affects to the performance of peer-to-peer networks because of two facts. First of all we have the service disruption because of handovers. Depending on the type of handover, macro-mobility or micro-mobility based, this time will be different and will affect in a major or minor way to the performance of the overlay. Although mobility protocols try to minimise this effect, typically we will always have a certain level of impact of the handovers in the performance. Furthermore, we have to take into account another fact, depending on the mobility solution a change of IP address can be needed when the terminal moves. For example if a terminal uses Mobile IP but it wants to register the CoA instead of the HoA in the DHT peer-to-peer network to avoid routing inefficiencies of using the HoA. Therefore, a modification in the maintenance algorithm of the DHT needs to be considered. This implies that the overlay routing tables have to be updated more frequently, and the maintenance traffic needed to update these overlay routing tables will also increase. If in addition to this problem, we consider that mobile nodes usually have limited bandwidth capabilities, the increment in the maintenance traffic does not seem to be a good solution. Furthermore, the mobile IP handovers also

introduce disruptions in the connectivity, these disruptions increment the churn suffered by the peer-to-peer overlay. Therefore, it would be desirable to minimize these effects as much as possible.

A. Management of routing tables in peer-to-peer overlay networks

Different methods to update the routing tables in DHTs have been proposed until this moment. In [9] two approaches are explained, they are proactive maintenance and reactive maintenance. In the first one, maintenance operations are run periodically in order to assure fresh routing entries and to avoid failures as much as possible. In the other approach called reactive maintenance, it fixes the errors once they are detected. Furthermore, many tweaks can be used on both approaches to improve the overall performance. The first approach is interesting for scenarios with high churn because the traffic generated to update the routing tables is limited by the periodicity that is used to refresh the entries. On the other hand, the second approach is suitable for scenarios where the churn is low. Only maintenance traffic is generated if necessary, and the errors caused are minimal because they don't occur frequently.

Finally, when some peers have a very high churn, it is better that they don't participate in the maintenance of the overlay. Its churn will produce more than drawbacks than the benefits of their resources to the overlay. The solution is to allow these peers to use the overlay but not to participate in its maintenance [13]. In P2PSIP this peers are called clients [4].

B. Management of peer-to-peer routing tables in mobile environments

The question that is discussed in this section is which is the most suitable strategy that must adopt a peer-to-peer overlay network if it is not desired to reduce the performance in a heterogeneous scenario with mobile peers. Several considerations can be done. One could consider using the approach of using the client profile for mobile nodes, so these peers wouldn't participate in the overlay [13]. However this approach cannot be applied in a scenario where only mobile peers exist. In this case it would be more suitable a proactive strategy in order to minimise the maintenance traffic of updating the overlay routing entries and avoiding as much as possible of the wireless interfaces of the peers. Nevertheless, in a heterogeneous scenario, stable peers will have to increase the costs of their maintenance traffic since mobile nodes exist, although a reactive strategy would be more suitable. Therefore, depending on the scenario one approach would be more suitable than other. Furthermore, we cannot predict how new services will evolve and which strategy would be the best.

We advocate for a flexible solution that can be adopted in any scenario. A classification of the different nodes participating in a peer-to-peer network can be done. One classification according to their mobility can be done as follows:

- **Fixed Nodes**
  - *Stable Nodes*: These nodes present large up-times and a stable connectivity. This fact usually implies a fixed available bandwidth and RTT in the access network.
  - *Unstable Peers*: These peers present small up-times. This behaviour is usually because of connectivity problems or own system instability. Bandwidth and RTT are usually stable but only available in short periods of time.
- **Mobile Nodes**
  - *Low Mobility Peers*: This profile considers those peers that have mobility support but they don't change their location very frequently. Although the bandwidth and RTT are given by the access network, they depend on the number of users that are connected in a cell or access point.
  - *High Mobility Peers*: These peers usually change their cell or visiting network since they change their location really fast. This pattern implies a lot of disruptions. Therefore, the RTT and bandwidth are heterogeneous and difficult to predict because of the continuous changes.

A different peer-to-peer overlay can be built according to the different groups listed before, and the most suitable strategy or DHT overlay [8] can be used. For fixed nodes we can use a reactive strategy, but for Unstable and Low Mobility Peers, both profiles with a higher churn, we can use a reactive algorithm tweaked to each of these profiles. Finally, high mobility peers can be configured as clients that are attached to the overlays maintained by the other profiles. Thus the problem that arises is how to allow the communication between the different overlays. This problem can be solved with H-P2PSIP [21] and [22] if we do an intelligent mapping of the different overlays in this architecture. Furthermore, this solution gives a great flexibility than can be really interesting for future deployments. The main drawback than can be related with this solution is the fact that probably is not a very good idea to have only mobility peers in an overlay network because their lifetimes probably would be short and the stability of super-peers peers could be affected in a dramatic way. This last statement depends on the strategies adopted for that profile and scalability of the solution but more stable peers can be also introduced but they will find drawbacks because they are not attached on their original overlay. Therefore some type of incentive mechanism is needed. Incentives in peer-to-peer systems is an open topic and it is out of scope with the topic of this paper, so it is not analysed but it will be probably be considered in an implementation.

VI. Hierarchical P2PSIP in mobile environments

The solution we propose is to change how resources are stored in H-P2PSIP ([21], [22]) in order to maintain different overlays of the same domain that manage peers with different mobility profiles. Using this approach, each peer-to-peer network can be optimised according to the specific node behaviour. Furthermore, in order to allow the connectivity between peers of different behaviours, a Hierarchical DHT based on P2PSIP [22] can be deployed to interconnect overlays with different peer behaviours; an example is in Figure 1. The peers in the same overlay share the same mobility profile and the connectivity between peers with different profiles is allowed through the interconnection overlay.

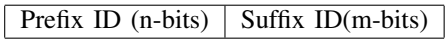


Fig. 3. Hierarchical ID

A. Hierarchical space domain of identifiers

In order to support the H-P2PSIP architecture, we define a hierarchical space of identifiers containing Hierarchical IDs (see Figure 3). Each Hierarchical ID is composed by two part IDs: a Prefix ID with n bits and a Suffix ID with m bits. The Prefix ID is used for the routing in the Interconnection Overlay between the different P2PSIP domains, whereas the Suffix ID is used for routing queries only in the own P2PSIP domain of a peer. This design advocates for a variable length for Node IDs in P2PSIP since any mapping function with independence of its length can be used to generate the Hierarchical ID. This Hierarchical ID can be used either as Node ID or Resource ID.

As Node ID identifies each node participating in the overlay network. The generation of the Node ID depends on the security level desired in the system. The simplest approach is to generate the Prefix Id of the node as `Prefix-ID=hash(domain_name.com)` and the Suffix ID as `Suffix-ID=hash(ip_address)`. However, if a more secured infrastructure wants to be provided, a central authority can be used to generate the certificates of a domain [23]. This central authority must define a mechanism to generate a Prefix ID per domain and a Suffix ID per peer in a domain, random numbers is a good approach to avoid some attacks to the overlay [23].

The generation of a Resource ID depends on the type of resource and how it is identified in the real world. This knowledge is necessary to accommodate its identification in the key space of a DHT. In our previous work [21] and [22], users and services are identified by URI's, like in a VoIP scenario based on SIP. The Prefix and Suffix IDs are generated with a hash of different parts of the URI. If we have a resource identified with the URI `resource@example.com`, we obtain:

- `Prefix-ID=hash(example.com)`
- `Suffix-ID=hash_a(resource@example.com)`

In this way all the resources of the same domain get the same Prefix ID and the Suffix ID identifies the resource. However, more complex and secured mapping functions can be used if necessary.

B. Peer mobility behaviours mapping

Considering our previous work [21] and [22], where users and services are identified by URI's, we have to incorporate the information of the peer mobility profiles in the URI format. The solution that has been adopted is to use a tag at the end of the URI that differentiates the mobility profile where is attached a peer according to its behaviour. The defined format is as follows: *user@example.org:**xx***. The *xx* tag defines where a user is attached and this tag can be *st* (**st**able peer), *un* (**un**stable peer), *lm* (**l**ow **m**obility peer) and *hm* (**h**igh **m**obility peer).

URI's are mapped to the Hierarchical ID in the following manner, the Prefix ID is obtained by applying a hash to the domain of the URI and the profile tag: `Prefix-ID=hash(example.com:xx)`. The Suffix ID is obtained from the hash of the URI without the profile tag: `Suffix-ID=hash_a(resource@example.com)`. The hash functions `hash` and `hash_a` can be identical or different. If something wants to be stored in the overlay network each Resource ID will have a Hierarchical ID format and it will have associated the original URI and the resource information. Each resource would be placed on the peer with the closest Node ID. Depending on the DHT protocol, this tuple can be replicated to other peers in some way. The content of the resource information can vary depending on the application scenario (i.e. location information in VoIP).

C. H-P2PSIP Basic Operation

Once the resources have been mapped to identifiers and how to storage them in the overlay, H-P2PSIP defines a method to locate these resources. This method is divided in two cases. In the first case, the search of a resource is bounded to the P2PSIP domain of the requester. This case is really simple since the search for resources is done inside the P2PSIP domain and it is identical to the flat peer-to-peer overlay using only the Suffix ID. In this situation, the Prefix ID of the resource must be equal to the hash of the associated URI domain. This hash is known by all the peers belonging to that P2PSIP domain. However, if a resource is stored in a different domain or in the same domain with a different mobility profile, the operation is more complex. For instance, this case can correspond to a VoIP call from a user in a P2PSIP domain to another user in a different P2PSIP domain. In order to obtain the resource (e.g. location) of the desired user, it is necessary to obtain the contact information published in the other P2PSIP domain. The first step in the search is to find a peer that can request information from other P2PSIP domains. These are the super-peers and there are several mechanisms [24], [25] that can be used to select them, which can be integrated in the maintenance protocol of the DHT used in the domain. Each P2PSIP domain has at least one super-peer, although it is desirable to have several super-peers for redundancy and performance.

Since all the peers in a domain know at least one super-peer, they can send a query to the super-peer in one hop. When the super-peer receives the query, it will search in the Interconnection Overlay for any of the super-peers that are responsible for the target Prefix ID, and once this information is retrieved, the query is forwarded to one of these super-peers. When the super-peer of the destination P2PSIP domain receives the query, it forwards the query inside its domain. If the query reaches a peer that has the desired resource, then the peer replies in a way that is compliant with the P2PSIP protocol [4].

An example of the signalling on the proposed hierarchical scenario is shown in Fig. 4, this example can be applied to a VoIP or an Instant Messaging service. Several aspects are taken into account in order to understand the signalling flow. First of all, when the peer in `domain.com:st` requests the information of `user1@domain.com:lm`, the query in the Fetch message is plain text. Plain text is used since a peer in a domain does not have to know what hash function is used in the Interconnection Overlay and what hash function is used in other P2PSIP domains. Thus, the super-peer

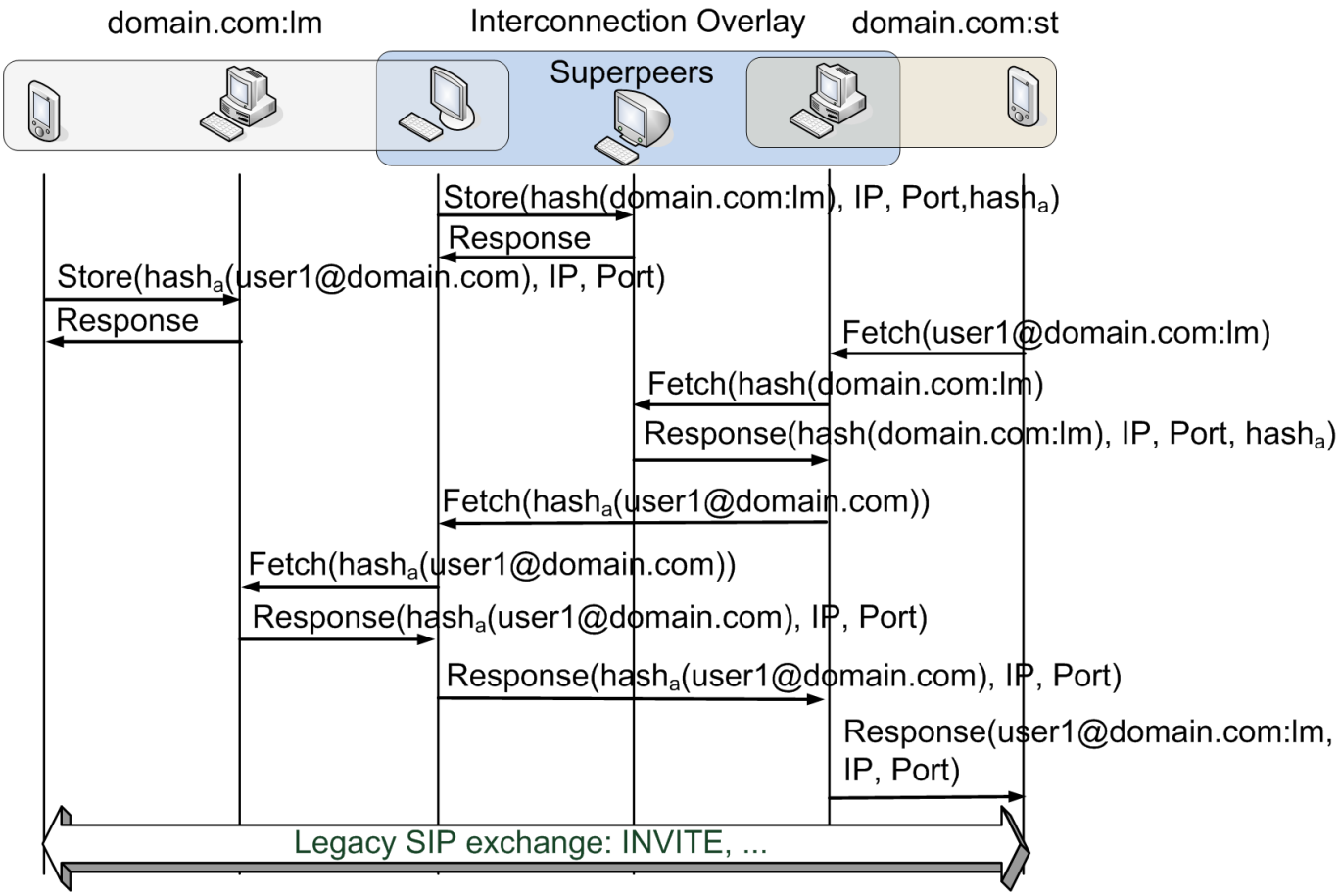


Fig. 4. H-P2PSIP Signalling

in `domain.com:st` performs `hash(domain.com:lm)` in order to obtain the information of the super-peers in `domain.com:lm` through the Interconnection Overlay. Inside this information, the hash used in the other domain ($hash_a$) is included and a request for the desired item can be built as `hash_a(user1@domain.com)`. Some of the peers taking care of the desired Resource ID answer to the super-peer from `domain.com:lm`, which then forwards this information to the super-peer from `domain.com:st`. Finally, the super-peer from `domain.com:st` sends the desired Resource ID to the peer from `domain.com:st`. Once this flow finishes, a SIP negotiation can be initiated for IM, VoIP or Video Conference. Figure 4 illustrates a subset of the real flow. The figure omits the intermediate hops in each overlay or Interactive Connectivity Establishment (ICE) exchanges for NAT traversal, if any is needed.

Therefore, the communication between different overlays is possible and different strategies can be adopted in each overlay. A different peer-to-peer overlay network can be used considering its robustness against churn or the stabilisation algorithm to update routing tables can be launched more frequently to compensate the churn effects.

D. Dynamic profile update

An important problem is how to contact with a peer with unknown mobility profile. One option could be to look in the last P2PSIP domain where it was contacted. If this information is not available, the first step is to look in the own domain where a peer is attached. Otherwise, each domain can be queried iteratively or in parallel. However, in order to avoid losing time and bandwidth with unnecessary queries, a peer can leave the information of its new position in the last visited domain. This information will be only available for a certain period of time. This solution is a compromise between looking for peers among all the domains and to store the location information in each one of the domains.

The way to proceed is as follows and it is illustrated on Figure 5. If a peer changes its location from the domain of low mobility to the domain of stable peers, it has to register this information in the new domain. Additionally, it has to register in the previous overlay domain a pointer to its new attachment point. In Figure 5, its URI with its new profile tag is stored on the original domain. If a peer looks for it in the old domain, it obtains the pointer of its new peer-to-peer location. Thus, it can start the same signalling exchange as explained in Figure 4 to get its contact location. Once these actions have been performed, a legacy SIP exchange can be done between the partners of the new session.

VII. Conclusions

In this paper, we propose an extension for P2PSIP domains that considers the mobility of peers. Due to the mobility environment, performance will be lower that the estimated in those references, but there are mechanisms to keep good

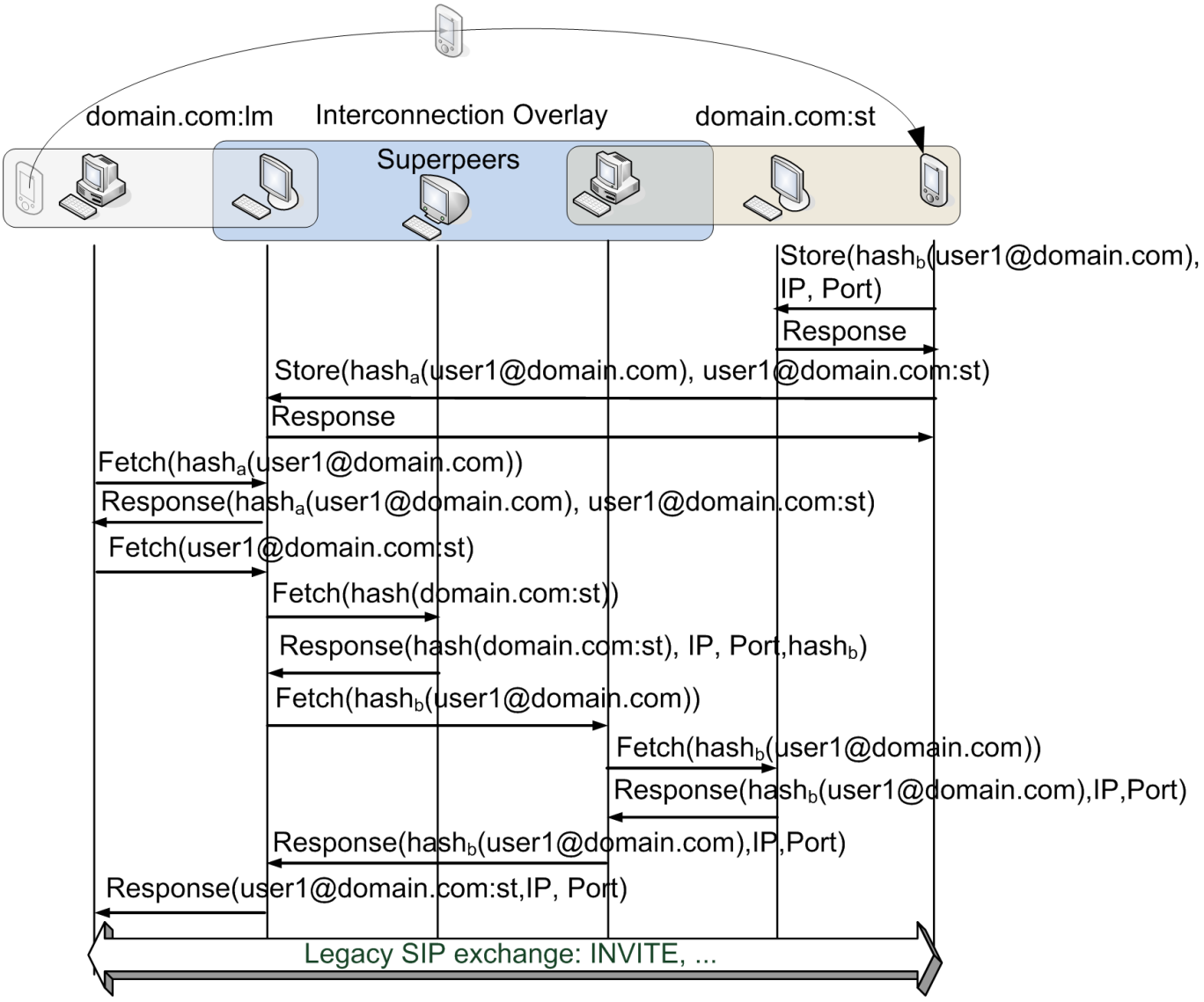


Fig. 5. Dynamic Update Signalling

levels of performance under mobility scenarios (i.e. [9], [13]). In this paper we argue that these should apply selectively only to the peers with high churn, in this way avoiding their negative impact on the rest of the peers and obtaining a better performance. Therefore, our proposal allows defining different domains for peers having different mobility behaviours. The connectivity between the different domains is realised through an interconnection overlay. In order to allow the routing between the different overlays through the interconnection overlay, a mapping function between the peers' URI and the tag that defines the mobility profile is defined to create the Hierarchical ID in a very simple way and with minor changes in comparison with our previous proposals [21], [22]. The Hierarchical ID is composed by a Prefix ID used for the routing in the interconnection overlay and a Suffix ID used in each overlay domain. The performance of this architecture without mobility considerations has been proved in [21] or [22].

The problem that arises under this architecture is to find the most suitable overlays and their setup parameters depending on the scenario and the mobile profile under study. A starting point can be [8] where the performance of different DHT's under churn is studied. Therefore, the next step in our research is to perform an evaluation of the benefits of this solution and its costs taking into account different type of mobility mechanisms, peer-to-peer networks, setup parameters and peer-to-peer maintenance solutions.

ACKNOWLEDGEMENT

We would like to acknowledge Ignacio Soto for their insightful comments to improve the readability and understanding of this paper.

This research work is being supported by the European Commission under the IST Content Network of Excellence[3] (FP6-2006-IST-038423), by the Regional Government of Madrid under the BioGridNet[4] project (CAM, S-0505/TIC-0101) and by the Ministry of Science and Innovation under the CONPARTE project (MEC, TEC2007-67966-C03-03/TCM).

[3]http://www.ist-content.eu
[4]http://www.biogridnet.org

REFERENCIAS

[1] S. A. Baset and H. G. Schulzrinne, "An analysis of the skype peer-to-peer internet telephony protocol," *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, April 2006.

[2] D. Rossi, M. Melia, and M. Meo, "A detailed measurment of skype network traffic," in *In IPTPS 2008*, 2008.

[3] D. Bryan, P. Matthews, E. Shim, and D. Willis, "Concepts and terminology for peer to peer sip," July 2008, internet Draft draft-ietf-p2psip-concepts-02.txt.

[4] C. Jennings, B. Lowekamp, E. Rescorla, S. Baset, and H. Schulzrinne, "Resource location and discovery (reload)," July 2008, internet Draft draft-ietf-p2psip-reload-00.txt.

[5] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications," *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 11, no. 1, 2003.

[6] P. Maymounkov and D. Mazieres, *IPTPS 2002 Cambridge, MA, USA, March 7-8, 2002. Revised Papers*, ser. Lecture Notes in Computer Science. Springer, 2002, vol. 2429/2002, ch. Kademlia: A peer-to-peer information system based on the XOR metric, pp. 53–65. [Online]. Available: http://www.springerlink.com/content/2ekx2a76ptwd24qt/?p=d9a88bd0609d4ac3902f147d1c183345&pi=0

[7] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, "A scalable content-addressable network," in *SIGCOMM '01*. New York, NY, USA: ACM Press, 2001, pp. 161–172.

[8] J. Li, J. Stribling, R. Morris, M. Kaashoek, and T. Gil, "A performance vs. cost framework for evaluating dht design tradeoffs under churn," *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 1, pp. 225–236 vol. 1, March 2005.

[9] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz, "Handling churn in a dht," in *ATEC '04: Proceedings of the annual conference on USENIX Annual Technical Conference*. Berkeley, CA, USA: USENIX Association, 2004, pp. 10–10.

[10] M. Steiner, T. En-Najjary, and E. W. Biersack, "Exploiting kad: possible uses and misuses," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 5, pp. 65–70, 2007.

[11] M. Steiner, T. En Najjary, and E. W. Biersack, "Analyzing peer behavior in KAD," Institut Eurecom, France, Tech. Rep. EURECOM+2358, Oct 2007.

[12] D. Stutzbach and R. Rejaie, "Understanding churn in peer-to-peer networks," in *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM, 2006.

[13] A. MacQuire, A. Brampton, I. Rai, and L. Mathy, "Performance analysis of stealth dht with mobile nodes," *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, pp. 5 pp.–189, March 2006.

[14] R. Aguiar, S. Sargento, A. Banchs, C. Bernardo, M. Calderon, I. Soto, M. Liebsch, T. Melia, and P. Pacyna, "Scalable qos-aware mobility for future mobile operators," *Communications Magazine, IEEE*, vol. 44, no. 6, pp. 95–102, June 2006.

[15] W. Kellerer, Z. Despotovic, M. Michel, Q. Hofstatter, and S. Zols, "Towards a mobile peer-to-peer service platform," Jan. 2007, pp. 2–2.

[16] O. Landsiedel, S. Gotz, and K. Wehrle, "Towards scalable mobility in distributed hash tables," Sept. 2006, pp. 203–209.

[17] A. T. Campbell and J. Gomez-Castellanos, "Comparison of ip micro-mobility protocols," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 4, no. 4, pp. 45–53, 2000.

[18] H. Soliman, C. Castelluccia, K. E. Malki, and L. Bellier, "Hierarchical Mobile IPv6 Mobility Management (HMIPv6)," Internet Engineering Task Force, RFC 4140, Aug. 2005. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4140.txt

[19] C. Perkins, "IP Mobility Support for IPv4," Internet Engineering Task Force, RFC 3220, Jan. 2002. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3220.txt

[20] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," Internet Engineering Task Force, RFC 3775, Jun. 2004. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3775.txt

[21] I. Martinez-Yelmo, A. Bikfalvi, C. Guerrero, R. Cuevas, and A. Mauthe, "Enabling global multimedia distributed services based on hierarchical dht overlay networks," in *NGMAST 2008. Future Multimedia Networking Workshop*. IEEE Computer Society, Sep. 2008, pp. 543–549.

[22] I. Martinez-Yelmo, A. Bikfalvi, R. Cuevas, C. Guerrero, and J. Garcia, "H-p2psip: Interconnection of p2psip domains for global multimedia services based on a hierarchical dht overlay network," *Computer Networks. (Accepted to Appear on March)*, vol. Special Issue on Content Distribution Infrastructures for Community Networks, 2009.

[23] G. Urdaneta, G. Pierre, and M. van Steen, "A survey of DHT security techniques," *ACM Computing Surveys*, 2009, http://www.globule.org/publi/SDST_acmcs2009.html, to appear.

[24] S.-H. Min, J. Holliday, and D.-S. Cho, "Optimal super-peer selection for large-scale p2p system," in *Hybrid Information Technology, 2006. ICHIT'06. Vol 2. International Conference on*, vol. 2, 2006.

[25] A. T. Mizrak, Y. Cheng, V. Kumar, and S. Savage, "Structured superpeers: leveraging heterogeneity to provide constant-time lookup," in *Internet Applications. WIAPP 2003. Proceedings.*, 2003, pp. 104–111.

Despliegue semántico de servicios en entornos heterogéneos y distribuidos

Laura Díaz-Casillas, Mercedes Garijo
Departamento de Ingeniería de Servicios Telemáticos
Universidad Politécnica de Madrid
Avenida Complutense 30, Madrid (España)
{ldcasillas, mga}@dit.upm.es

***Resumen*—En la actualidad, numerosas aplicaciones empresariales se desarrollan siguiendo una arquitectura orientada a servicios. Una de las actividades asociadas a este proceso es el despliegue, el cual debe realizarse sobre varios entornos, adecuados a las necesidades de cada fase del desarrollo. Dichos entornos se caracterizan por ser heterogéneos y distribuidos. En la mayor parte de los casos, el despliegue se realiza de forma manual, siendo una tarea costosa y que genera numerosos errores, existiendo una necesidad de encontrar alguna solución que facilite dicha labor. En este artículo, se propone emplear una ontología mediante la cual se caractericen los elementos involucrados en el despliegue de servicios, considerando sus propiedades funcionales y no funcionales, con el objetivo de realizar un despliegue óptimo, adecuado a las características del entorno de desarrollo concreto sobre el que se trabaja.**

***Palabras Clave*—ontología despliegue servicios**

I. INTRODUCCIÓN

El uso de servicios web [1] y arquitecturas orientadas a servicios (o SOA, del inglés *Service Oriented Architecture*) [2] favorece el desarrollo de aplicaciones empresariales, facilitando la interoperabilidad entre sistemas heterogéneos, al definir una manera estándar de anunciar e invocar servicios que actúan de forma independiente, mediante los cuales se alcanzan los objetivos de negocio deseados. Su uso fomenta además la disponibilidad, fiabilidad y escalabilidad de dichas aplicaciones, lo que ha llevado a su implantación en un gran número de empresas.

Una de las actividades a llevar a cabo durante el desarrollo de este tipo de aplicaciones es el despliegue de servicios, la cual consiste en realizar todas las acciones necesarias para poder poner los servicios en funcionamiento. Un servicio es un sistema software capaz de proporcionar una función a un usuario o a otro sistema software. Es habitual que los servicios cooperen entre sí para poder alcanzar el objetivo deseado, siendo necesario comprobar su disponibilidad. Por otro lado, también se deben analizar los requisitos demandados por los servicios para su correcta ejecución, y seleccionar en base a éstos los nodos más adecuados para su despliegue. Es además necesario tener en cuenta que durante el desarrollo de los servicios es habitual que se produzcan modificaciones en sus requisitos de despliegue, variándose las características de los componentes requeridos para su correcta ejecución, lo que da lugar a que las actividades de despliegue cambien a lo largo del desarrollo.

Pero aparte de las requisitos específicos del servicio que se desea poner en funcionamiento, es necesario considerar las características del entorno sobre el que se va a realizar el despliegue. En concreto, los entornos de despliegue de aplicaciones empresariales se caracterizan por ser heterogéneos, estando constituidos por componentes de diversas características, y fuertemente distribuidos, es decir, los servicios suelen ejecutarse sobre distintos nodos, los cuales se encuentran conectados mediante enlaces de red. Por otro lado, se debe tener en cuenta que durante el desarrollo de las aplicaciones empresariales se suelen utilizar varios entornos de despliegue, para de esta forma adecuarse a las necesidades concretas de cada fase del proceso de desarrollo. Así, el número de recursos empleados en las fases iniciales varía con respecto a los utilizados en producción.

La complejidad de las actuales aplicaciones empresariales hace que el despliegue sea una actividad complicada. Sin embargo, es una tarea que se realiza normalmente de forma manual, siendo por ello una de las principales fuentes de error durante el desarrollo de los sistemas, y que por tanto presenta un alto coste asociado. Este hecho ha dado lugar a la investigación en este campo, con el objetivo de facilitar la realización de dicha tarea, proponiéndose diversas soluciones para poder llevar a cabo el despliegue de los servicios en distintos entornos. En concreto, el despliegue puede realizarse de forma manual o a través de herramientas basadas en *scripts*, lenguajes o modelos. En [3] se realiza un análisis de dichas opciones, llegándose a la conclusión de que la opción manual es la que presenta menos barreras de entrada y mayor facilidad de uso, pero se encuentra muy limitada a la hora de escalar y realizar modificaciones en el sistema, recomendándose las técnicas basadas en modelos.

En este artículo se propone el uso de tecnologías semánticas para realizar el despliegue de servicios de aplicaciones empresariales sobre entornos heterogéneos y distribuidos. En concreto, se propone emplear una ontología mediante la cual modelar los elementos involucrados en dicho proceso, es decir, tanto los recursos físicos como los recursos lógicos. En ella, se tendrán en cuenta requisitos funcionales y no funcionales, para llevar a cabo el proceso de despliegue de la forma más adecuada posible al entorno sobre el que se trabaje.

El resto del artículo se estructura de la siguiente manera. En el capítulo II se muestra un escenario de ejemplo, en el cual se explica una posible aplicación del despliegue semántico de los servicios. En el capítulo III se realiza un estudio previo de las tecnologías empleadas en la solución propuesta, descrita en el capítulo IV. A continuación, en el capítulo V se presentan trabajos relacionados y por último, en el capítulo VI, se exponen las conclusiones y líneas futuras de trabajo.

II. Escenario de ejemplo

Se expone a continuación un escenario de aplicación, enmarcado dentro del desarrollo del proyecto ITECBAN (Infraestructura Tecnológica y Metodológica de Soporte para un Core Bancario), cuyo principal objetivo es la elaboración de una plataforma que sirva como base para la creación de sistemas de gestión destinados al sector bancario, eliminándose las actuales limitaciones de los sistemas de información empleados en entornos financieros.

Las aplicaciones empresariales empleadas en este tipo de áreas se caracterizan por presentar en la mayor parte de los casos una arquitectura orientada a servicios, lo que las permite adaptarse de manera continua y flexible a los cambios que se producen en su alrededor, en respuesta a la demanda del mercado. Durante el desarrollo de este tipo de aplicaciones es necesario el empleo de varios entornos de despliegue, a través de los cuales se evoluciona hasta alcanzar el entorno de producción. Los motivos que conducen a utilizar distintos entornos son diversos, entre ellos destacan el control de acceso, el cual debe ir aumentando a medida que se avanza en el proceso de desarrollo, y la topología del entorno, la cual va cambiando a lo largo del ciclo de desarrollo, haciéndose más compleja en la fase de producción, cuando se requiere ampliar el número de recursos físicos empleados, siendo necesario utilizar sistemas de respaldo. A esta situación se añade el hecho de que dichos entornos son heterogéneos y distribuidos, siendo frecuente que se produzcan cambios en sus topologías, muchas veces debidos al mantenimiento o a la mejora de sus prestaciones.

Surge entonces la necesidad de disponer de una solución que facilite el despliegue de los servicios, adecuándolos a las necesidades y características de cada entorno en concreto. Para ello, se propone utilizar una ontología mediante la cual modelar los recursos presentes en el entorno, y así poder determinar los nodos más adecuados sobre los que desplegar cada uno de los servicios.

En la base de conocimiento de la ontología se almacenará la información de los recursos disponibles en el sistema, para en base a la selección del usuario de los servicios a desplegar sobre un entorno concreto, analizar las necesidades demandadas por éstos junto con las características del entorno sobre el que se desea realizar el despliegue, y así poder determinar la ubicación más adecuada de los servicios y conseguir que éstos funcionen correctamente.

III. Estudio previo

Una ontología es una representación formal de los conceptos pertenecientes a un dominio concreto, en la que se indican sus características y las relaciones entre dichos conceptos.

La definición de los elementos presentes en un dominio mediante una ontología facilita el intercambio de información, permitiendo la integración de datos procedentes de diversas fuentes, como son modelos de información, servicios o comportamiento, posiblemente representados en distintos lenguajes, al pertenecer a sistemas diferentes. Por otro lado, una ontología se caracteriza por ser una representación del conocimiento, permitiendo implementar capacidades de razonamiento sobre ella.

Pero además, la aplicación de ontologías al despliegue de servicios aporta diversas ventajas:

- La representación de los conceptos asociados al despliegue de servicios mediante una ontología posibilita su evolución, añadiendo nuevos elementos e incluso adaptando las propiedades de los elementos existentes a las nuevas características que puedan aparecer en un futuro, mientras no se varíe el modelo original, facilitando con ello la reutilización de conceptos existentes.
- El uso de una ontología permite realizar una verificación del modelo, comprobando su consistencia y detectando redundancias, evitando con ello posibles errores antes de realizar las actividades asociadas al proceso de despliegue.
- La descripción semántica de los recursos empleados facilita su búsqueda, favoreciendo la selección de los más apropiados a cada caso.
- Una ontología representa la información de manera jerárquica, lo que facilita su interpretación al mostrarse de forma más clara las asociaciones y dependencias entre los elementos involucrados en el despliegue que si la representación se realizara de manera textual.

En [4] se analiza cómo el uso de ontologías en sistemas de gestión permite representar conceptos, tales como conjuntos, relaciones de similitud o enlaces entre distintas representaciones de una entidad, que no pueden implementarse con otras tecnologías empleadas en la actualidad para modelar este tipo de sistemas, como es por ejemplo UML (*Unified Modeling Language*).

El lenguaje de ontologías web (u OWL, del inglés *Web Ontology Language*) [5] es un lenguaje de marcado que permite definir clases, mediante las cuales representar los conceptos, y sus propiedades asociadas, a través de las cuales determinar las relaciones y restricciones existentes entre ellas. De este modo, OWL posibilita la construcción de una ontología en la que se modelen todos los elementos partícipes en el despliegue de los servicios.

Las posibilidades de OWL pueden ampliarse mediante SWRL (*Semantic Web Rule Language*) [6], un lenguaje propuesto por el W3C (*World Wide Web Consortium*) que permite definir reglas y combinarlas con la base de conocimiento de OWL. Las reglas proporcionadas se corresponden con cláusulas de Horn, estableciendo una relación entre un antecedente y un consecuente, de manera que cuando se cumplan las condiciones expuestas en el antecedente, las condiciones indicadas en el consecuente deberán cumplirse también. De esta forma, se consigue enriquecer el modelo representado mediante OWL a través de una serie de reglas que ayudan a determinar las actividades a realizar durante el proceso de despliegue.

IV. Ontología propuesta

A continuación, se describe la ontología propuesta, mediante la cual se pretende representar los recursos involucrados en el despliegue de servicios, incluyéndose sus propiedades y las relaciones entre ellos, con el objetivo de facilitar dicho proceso, adecuándolo a las características de cada caso concreto.

Considerando la rápida evolución de los sistemas presentes en la industria, se ha optado por proponer una ontología básica, en la que se consideran las características más representa-

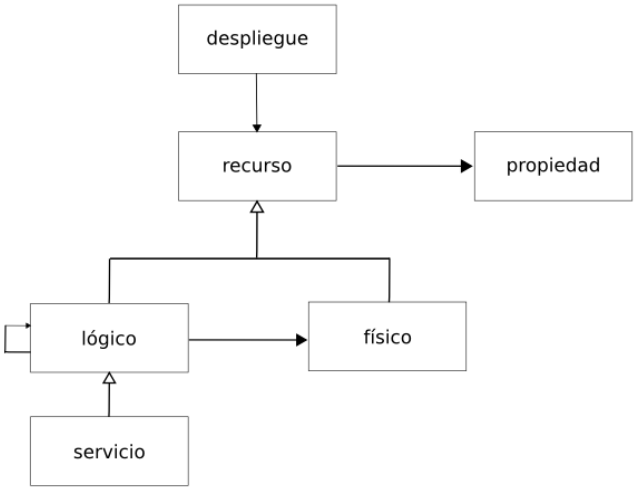


Figura 1. Despliegue del sistema

tivas de los elementos partícipes del despliegue, facilitándose su ampliación y adaptación a necesidades futuras.

Si se analizan los elementos que componen un entorno de despliegue, se observa como éste se encuentra formado por una serie de recursos, físicos y lógicos. Los recursos físicos representan los nodos sobre los que se realizará el despliegue, mientras que los recursos lógicos identifican los sistemas software a desplegar, entre los cuales están los servicios. Los recursos lógicos se distinguen además porque pueden requerir otros recursos lógicos para poder funcionar.

Por otro lado, es necesario establecer una relación directa entre los recursos lógicos y físicos, debido a que cada recurso lógico se desplegará sobre un determinado recurso físico, dotando a éste de un valor añadido, que en algunos casos determinará el despliegue de otros recursos que dependan de él.

Cada uno de los recursos se encuentra identificado por medio de un nombre único y definido mediante una serie de propiedades, caracterizadas a través de uno o más atributos, que llevan asociada una medida para poder cuantificarlos.

Esta información queda reflejada en la figura 1, en la que se muestra una visión general de la ontología propuesta.

IV-A. Descripción de las propiedades de los recursos

Los recursos presentan una serie de propiedades funcionales y no funcionales que los definen.

Cada propiedad se encuentra caracterizada mediante un nombre y uno o más atributos, los cuales a su vez tienen asociada una medida, que permite valorarlos de manera objetiva o subjetiva, mediante un valor y un tipo que determina cómo tratar dicho valor.

Si las propiedades funcionales del nodo se corresponden con las demandadas por los servicios a desplegar sobre él, se consigue una correcta ejecución de los servicios, pero si además se tienen en cuenta características no funcionales, como la disponibilidad o la seguridad de los servicios, se obtiene un despliegue más eficiente y adecuado a las características del entorno de despliegue concreto sobre el que se actúa. Es por ello que se han considerado ambas a la hora de caracterizar los recursos en la ontología.

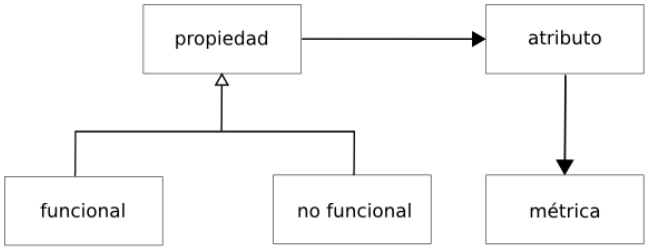


Figura 2. Propiedades

IV-A1. Propiedades funcionales: Para poder realizar un despliegue válido de los servicios es necesario analizar en primer lugar las propiedades funcionales de los recursos involucrados en el proceso. Tras el estudio de las necesidades demandadas por los servicios a desplegar, es necesario conocer los recursos disponibles en el entorno sobre el cual tendrá lugar el proceso, de manera que los servicios se alberguen en los nodos adecuados. Para lograr este propósito, se han considerado cuatro propiedades funcionales:

- Capacidad de procesamiento.
- Disco duro.
- Memoria.
- Conexión en red.

IV-A2. Propiedades no funcionales: Además, en la ontología se han tenido en cuenta propiedades no funcionales, con el objetivo de obtener una calidad de servicio adecuada a la demandada por la aplicación sobre el entorno de despliegue empleado. Teniendo en cuenta que éstos se caracterizan por ser heterogéneos y distribuidos, los aspectos que se han considerado más relevantes son:

- Coste asociado al despliegue del servicio, lo que dependerá del número de recursos requerido para su puesta en funcionamiento y de si éstos se encuentran actualmente en el sistema o es necesario desplegarlos.
- Disponibilidad o probabilidad de que un servicio responda a las peticiones de los clientes, lo que se encuentra directamente relacionado con la capacidad de respuesta del servicio frente a dichas peticiones.
- Fiabilidad o probabilidad de que un servicio se ejecute correctamente, lo que dependerá de las operaciones que realice, si éstas son locales o remotas o si dependen de otros servicios.
- Seguridad, en función de si un servicio requiere autenticación para su uso y codifica los datos que maneja o no.
- Interoperabilidad o facilidad que presenta un servicio para comunicarse con otros.
- Escalabilidad o capacidad de ampliación del servicio, en función de los recursos requeridos para su ejecución.

En la figura 2 se puede observar una visión más detallada de la parte de la ontología dedicada a la definición de las propiedades de los recursos.

IV-B. Descripción de los tipos de recursos

Por otro lado, los recursos se clasifican en dos grandes grupos: físicos y lógicos.

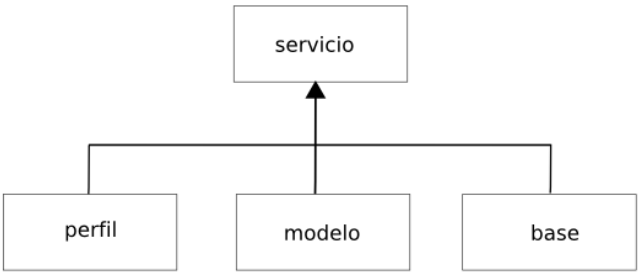


Figura 3. Servicios

IV-B1. Recursos físicos: Los recursos físicos se corresponden con los nodos sobre los cuales se despliegan los servicios. Éstos se caracterizan por tener una ubicación concreta, por tanto deben tener asociados aparte de un nombre, una dirección IP que permita localizarlos.

IV-B2. Recursos lógicos: Los recursos lógicos definen los recursos software a desplegar o necesarios para el despliegue de determinados recursos. Así por ejemplo, para el despliegue de un servicio que gestione una base de datos, será necesario instalar un sistema de gestión de bases de datos en el nodo correspondiente.

Los recursos lógicos se encuentran caracterizados por un nombre único, aunque además es importante indicar su estado: activo o inactivo, según hayan sido desplegados o no en el sistema. Por otro lado, también es necesario indicar su versión para evitar posibles incompatibilidades y facilitar su gestión durante las fases iniciales de desarrollo [7].

Dentro de los recursos lógicos se encuentran los servicios, descritos mediante OWL-S [8]. Ésta es una ontología definida por el W3C que permite incorporar semántica a los servicios web. Su objetivo es automatizar las tareas de descubrimiento, invocación, composición e interoperabilidad de servicios. En OWL-S, los servicios se describen mediante tres tipos de conocimiento:

- El perfil de servicio (*service profile*), que contiene una descripción semántica. Se emplea para anunciar el servicio, indicando su función a los posibles clientes.
- El modelo de servicio (*service model*), mediante el cual se define cómo se utiliza, indicando cómo debe invocarse el servicio y qué sucede durante su llamada.
- La base del servicio (*service grounding*), a través de la cual se define cómo se interacciona con él, proporcionando información acerca de los protocolos de comunicación, formatos de mensajes y otros detalles específicos que permiten conocer cómo acceder al servicio.

En la figura 3 se muestran los elementos que forman parte de OWL-S.

La descripción detallada que se realiza de los servicios al emplear OWL-S permite optimizar el proceso de despliegue. Al disponer de información detallada de cada uno de los servicios, es posible determinar con mayor precisión los nodos más adecuados para su despliegue. Además, el empleo de semántica para la descripción de los servicios permite la posibilidad de realizar un descubrimiento automático, y así por ejemplo, en el caso de que un servicio fallara, sería posible implementar un buscador semántico capaz de encontrar un servicio con características similares que lo sustituyera.

```
<!-- Service description-->

<service:Service rdf:ID="PersonalInformationService">
 <service:presents rdf:resource="#PersonalInformationProfile"/>
 <service:describedBy rdf:resource="#PersonalInformationProcess"/>
 <service:supports rdf:resource="#PersonalInformationGrounding"/>
</service:Service>
```

Figura 4. Descripción del servicio

```
<!-- Profile description -->

<profile:Profile rdf:ID="PersonalInformationProfile">
 <service:presentedBy rdf:resource="#PersonalInformationService"/>
 <service:has_process rdf:resource="#PersonalInformationProcess"/>
 <profile:serviceName xml:lang="en">Personal information</profile:serviceName>
 <profile:textDescription xml:lang="en"> This service returns the information
 of a person whose ID is given.
 </profile:textDescription>
 <profile:hasInput rdf:resource="#ID"/>
 <profile:hasOutput rdf:resource="#PersonalInformation"/>
</profile:Profile>
```

Figura 5. Perfil del servicio

A modo de ejemplo, en las figuras 4 y 5 se muestra parte de la descripción semántica de un servicio realizada mediante OWL-S, en la que se observa como asociados a cada servicio se definen un perfil, un modelo y una base. En el perfil se muestra el anuncio del servicio y los tipos de datos de entrada y salida. En este caso, el servicio proporciona la información de una persona a partir de su DNI.

IV-C. Validación

El uso de una ontología para describir los recursos permite realizar una validación de la información contenida en la base de conocimiento antes de llevar a cabo el proceso de despliegue, eliminando redundancias e incoherencias en los datos. De esta forma, se consiguen evitar posibles fallos en las actividades de despliegue.

Gracias a las restricciones indicadas en la definición de los recursos del sistema, es posible por ejemplo analizar las dependencias de los servicios a desplegar. Siendo necesario comprobar además que los recursos requeridos se encuentran disponibles en el sistema antes de realizar el despliegue, en caso contrario, será necesario ponerlos en funcionamiento.

IV-D. Restricciones adicionales

Se propone emplear SWRL como lenguaje de definición de reglas para complementar las restricciones indicadas mediante la ontología. SWRL permite definir reglas a través de las cuales analizar las propiedades de los recursos involucrados en el despliegue, adecuando la ubicación de los servicios a los nodos disponibles, para lograr un correcto funcionamiento. Como hemos visto en apartados anteriores, se consideran tanto las características funcionales como las no funcionales, con el fin de poder llevar a cabo un despliegue de servicios válido y adecuado al entorno de desarrollo sobre el que se trabaja.

El uso de SWRL permite, además de completar las restricciones impuestas por OWL, aplicar políticas mediante las cuales optimizar el proceso de despliegue. Por ejemplo, analizando los recursos empleados por un servicio y buscando otro servicio con la misma funcionalidad pero que requiera menos recursos. Además, también es posible la aplicación de políticas para realizar un reparto equitativo de carga entre los nodos disponibles o incluso dedicar nodos en exclusiva a determinados servicios o replicar un servicio en múltiples nodos de manera simultánea debido a las características de éste.

```
service (?a) ∧ node(?b) ∧ isLocated(?a, ?b)
⇒ swrlb:subtract(memory(?b), memory(?a))
```

Figura 6. Uso de SWRL I

```
service (?a) ∧ node(?b) ∧ swrlb:lessThan(memory(?a), memory(?b))
∧ node(?c) ∧  swrlb:lessThan(memory(?b), memory(?c))
⇒ deploy(?a, ?c)
```

Figura 7. Uso de SWRL II

En la figura 6 se muestra una regla implementada con SWRL, mediante la cual se permite añadir una restricción en la que el valor de una variable depende del valor de otra variable, limitación que no es posible implementar en OWL. En este caso, el servicio a está desplegado sobre el nodo b, por tanto, el valor de la memoria disponible en el nodo b depende de la memoria requerida por el servicio a. En el ejemplo de la figura 7, se emplea SWRL para definir una regla más compleja, cuyo objetivo es realizar un reparto equitativo del uso de memoria sobre los distintos nodos involucrados en el despliegue. En concreto, se compara la memoria disponible en los nodos b y c, realizándose el despliegue del servicio a sobre el nodo c, en el caso de que éste disponga de más memoria que el nodo b.

Se muestran sólo dos ejemplos del uso de SWRL en el proceso de despliegue de los servicios, siendo necesaria la aplicación de muchas otras reglas para poder obtener un resultado óptimo.

V. Trabajos relacionados

El despliegue de servicios en aplicaciones empresariales es una actividad compleja, que requiere tiempo y esfuerzo, y que normalmente se realiza de forma manual, siendo una de las principales fuentes de error y que por tanto más coste supone en el desarrollo de las aplicaciones.

Este hecho ha originado la propuesta de diversas ideas para solventar esta situación, dichas soluciones se encuentran basadas en *scripts*, lenguajes o modelos. Aunque la última opción es la que puede resultar más complicada en un principio, es también la que presenta mayores posibilidades a la hora de escalar el sistema y por ello es la empleada en este artículo. Pero existen otras propuestas en las que se emplean modelos para representar los elementos involucrados en el proceso de despliegue con el objetivo de facilitar las actividades requeridas para llevar a cabo dicho proceso, aunque ninguna de ellas se ha establecido como estándar. Así por ejemplo, en [9] se desarrolla un metamodelo mediante el cual se describen los parámetros de configuración requeridos para el despliegue de servicios, y en [10] se presenta otro metamodelo que trabaja a más alto nivel, centrado en la gestión de los servicios empleados por las aplicaciones, facilitando su compartición y reutilización. Los modelos suelen encontrarse integrados en sistemas más amplios, con el objetivo de automatizar las tareas de despliegue, como es por ejemplo [11], donde se propone una arquitectura de despliegue dirigida por políticas que permite adaptarse a entornos de despliegue distribuidos y heterogéneos, o [12], donde se define un entorno para el despliegue de servicios asociados a sesiones sobre un entorno distribuido. En dichas iniciativas se emplean modelos implementados con UML, lo que limita sus posibilidades.

Dicha limitación ha dado lugar a propuestas en las que se enriquecen los modelos mediante el uso de ontologías. Así en [13] se emplea OWL-S para describir los servicios, el entorno de despliegue y el propio proceso de despliegue, con el objetivo de obtener un resultado óptimo en entornos dinámicos. En [14] se propone una ontología extensible implementada en OWL que permite crear infraestructuras adaptables al contexto en el que se encuentren, adaptando las necesidades de los usuarios a las características concretas de los dispositivos y el entorno sobre el que se despliegan los servicios. En [15] y [16] se emplea OWL para definir ontologías mediante las cuales se modela el contexto para facilitar la autogestión de sistemas distribuidos orientados a objetos. En estos trabajos, se puede observar como el uso de ontologías mejora el proceso de despliegue, pero no tienen en cuenta las características no funcionales, con lo que limitan las posibilidades de obtener una despliegue óptimo.

VI. Conclusiones y trabajos futuros

El desarrollo de aplicaciones empresariales requiere el empleo de varios entornos de despliegue, heterogéneos y distribuidos, adecuados a las necesidades de cada fase del desarrollo. El despliegue es un proceso que se realiza normalmente de forma manual, dando lugar a numerosos errores, siendo necesario el desarrollo de una solución que facilite el correcto despliegue de los servicios sobre dichos entornos.

En este artículo, se propone una ontología mediante la cual se representan los recursos involucrados en el despliegue de los sistemas, considerándose tanto sus propiedades funcionales como funcionales, y sobre la que se aplican reglas, para obtener una correcta adecuación a las necesidades de cada entorno concreto.

Siguiendo el modelo de computación autonómica propuesto por IBM [17], como línea futura se propone ampliar la solución presentada para que el sistema sea capaz de reaccionar ante los cambios que se produzcan en el entorno tras su despliegue, determinando las acciones más adecuadas a realizar en cada caso para mantener el sistema en un estado óptimo de funcionamiento, solventando los posibles errores que puedan producirse.

Agradecimientos

Este trabajo ha sido cofinanciado por el Ministerio de Industria, a través del programa de Consorcios Estratégicos Nacionales en Investigación Técnica (CENIT) mediante el proyecto ITECBAN, y por el Ministerio de Ciencia e Innovación, a través del Plan Nacional de I+D+I, por medio del proyecto T2C2 (Tecnologías Telemáticas para la Colaboración Ciudadana - TIN2008-06739-C04-01).

Referencias

[1] W3C, “Web Services Activity,” http://www.w3.org/2002/ws/, 2009.
[2] S. Hashimi, “Service-Oriented Architecture Explained,” O’Reilly Media, Inc., 2003.
[3] V. Talwar, D. Milojicic, Q. Wu, C. Pu, W. Yan, and G. Jung, “Approaches for Service Deployment,” *IEEE Internet Computing - Service-Oriented Computing Track*, 2005.

[4] J. Strassner, D. O'Sullivan, and D. Lewis, "Ontologies in the Engineering of Management and Autonomic Systems: A Reality Check," *Journal of Network and Systems Management*, 2007.

[5] D. L. McGuinness and F. van Harmelen, "OWL Web Ontology Language Overview," W3C Recommendation, 2004.

[6] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML," W3C Member Submission, 2004.

[7] H. H. Shahri, J. A. Hendler, and A. A. Porter, "Software Configuration Management Using Ontologies," *3rd International Workshop on Semantic Web Enabled Software Engineering (SWESE 2007)*, 2007.

[8] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S.Ñarayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara, "OWL-S: Semantic Markup for Web Services," W3C Member Submission, 2004.

[9] Y. Li, J. Qiu, K. Sun, and Y. Chen, "Modeling and Verifying Configuration in Service Deployment," *IEEE International Conference on Services Computing (SCC'06)*, 2006.

[10] A. Chazalet and P. Lalanda, "A Meta-Model Approach for the Deployment of Services-oriented Applications," *IEEE International Conference on Services Computing (SCC' 2007)*, 2007.

[11] J. L. Revuelta, "A policy-driven, model-based software and services deployment architecture for heterogeneous environments," Departamento de Ingenieria de Sistemas Telematicos, Universidad Politecnica de Madrid, Tech. Rep., 2007.

[12] T. Villemur and E. Hammami, "Design and evaluation of a context-aware service deployment for collaborative sessions," *Computer Communications*, 2008.

[13] H. Chamekh and F. le Mouel, "An Ontology-based Approach to Semantically Deploy Services in Pervasive Environments," *IEEE International Conference on Pervasives Services*, 2007.

[14] D. Preuveneers, J. V. den Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx, Y. Berbers, K. Coninx, V. Jonckers, and K. D. Bosschere, "Towards an Extensible Context Ontology for Ambient Intelligence," in *Second European Symposium on Ambient Intelligence*, 2004.

[15] A. R. Haydarlou, M. A. Oey, B. J. Overeinder, and F. M. T. Brazier, "Using Semantic Web Technology for Self-Management of Distributed Object-Oriented Systems," *IEEE/WIC/ACM International Conference on Web Intelligence*, 2006.

[16] Y. Zhou, J. Pan, X. Ma, B. Luo, X. Tao, and J. Lu, "Applying ontology in architecture-based self-management applications," in *2007 ACM Symposium on Applied Computing*, 2007.

[17] J. O. Kephart and D. M. Chess, "The Vision of autonomic computing," *IEEE Computer Society*, 2003.

Modelo de seguridad para entornos colaborativos distribuidos y ubicuos y su aplicación a los NGCWE's

Jasone Astorga, Jon Matias, Eduardo Jacob
Departamento de Electrónica y Telecomunicaciones
Universidad del País Vasco / Euskal Herriko Unibertsitatea
Escuela Superior de Ingeniería. Alameda de Urquijo s/n. 48013 - Bilbao
jasone.astorga@ehu.es, jon.matias@ehu.es, eduardo.jacob@ehu.es

***Abstract*- Today, collaboration is considered a way to improve efficiency and quality of work and this concept is spreading quickly over the technological world, giving place to the creation of ubiquitous and distributed collaborative environments where the different high level services are composed from more basic modules. A typical example of this kind of environments are the NGCWEs (Next Generation Collaborative Working Environments), which are mainly characterized by the utilization of heterogeneous and low capacity mobile devices. In this paper we take into account all the special features that characterize NGCWEs in order to propose a security model which best fits their requirements. With this aim, we have developed a solution based on symmetric cryptography, and we have introduced the concept of a centralized authorization server, in order to minimize the load that the security solution imposes over the final systems.**

***Palabras Clave*- Autenticación, autorización, entornos distribuidos, Kerberos, seguridad en Servicios Web**

I. INTRODUCCIÓN

Este trabajo presenta una arquitectura de seguridad específicamente adaptada a entornos distribuidos en los que la gran mayoría de las entidades participantes son dispositivos móviles de capacidades reducidas. Un ejemplo típico de este tipo de entornos son los *Entornos de Trabajo Colaborativos de Nueva Generación*, comúnmente conocidos como NGCWEs (Next Generation Collaborative Working Environments) [1], los cuales se basan en promocionar la colaboración como medio para incrementar la eficiencia y la calidad del trabajo. En estos entornos los terminales cliente están constituidos a menudo por sensores, teléfonos móviles, PDAs, y otros dispositivos inalámbricos con capacidades limitadas.

La Fig. 1 muestra un ejemplo de la arquitectura básica de un entorno de este tipo. Esta arquitectura concreta es la que se ha especificado en el proyecto integrado C@R "A Collaborative Platform for Working and Living in Rural Areas" del VI Programa Marco, en cuyo marco se ha llevado a cabo el trabajo aquí presentado. Tal y como se puede ver en la Fig. 1, los entornos NGCWE están compuestos por un conjunto de elementos básicos y una serie de herramientas colaborativas o *middleware*. Los primeros son los componentes individuales asociados a las funcionalidades individuales de los diferentes entornos, es decir, son las piezas básicas utilizadas para construir los entornos de trabajo colaborativos. Estos elementos básicos pueden ser agrupados en diferentes bloques, tales como servicios de acceso (WiFi, WiMAX, GPRS, UMTS), servicios de red (IPv4, IPv6), mecanismos de autenticación y autorización (biométricos, usuario-contraseña, certificados), etc. Las herramientas colaborativas, por su parte, se definen como mecanismos que permiten integrar los componentes individuales de los CWEs. Estas herramientas facilitan la cooperación entre los diferentes elementos básicos, produciendo un resultado muy superior a la suma de las funcionalidades individuales. En otras palabras, las herramientas colaborativas permiten la creación dinámica y bajo demanda de aplicaciones cooperativas de alto nivel.

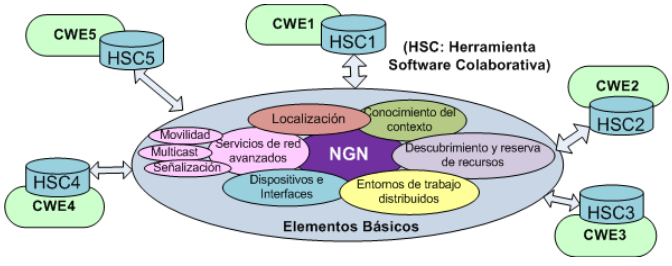


Fig. 1. Ejemplo de la arquitectura básica de un entorno NGCWE

Teniendo en cuenta las características de los NGCWEs recién descritas, resulta obvio que un sistema distribuido, muti-usuario y multi-tecnología como el presentado puede ser objeto de múltiples ataques de seguridad, entre los cuales caben destacar los ataques de seguridad propios de los sistemas de compartición de recursos: acceso no autenticado y/o no autorizado a datos y recursos, intercepción de sesiones, ataques de repetición, ataques de denegación de servicio (DoS), etc. Por este motivo, el objetivo de nuestro trabajo es proporcionar a los usuarios de este tipo de entornos una solución de seguridad que les permita establecer comunicaciones autenticadas, autorizadas y confidenciales entre las diferentes entidades que componen las aplicaciones colaborativas.

Uno de los aspectos fundamentales de este tipo de entornos es que requieren acceso transparente y ubicuo, de tal forma que los NGCWEs estén disponibles en cualquier momento y desde cualquier lugar o tipo de terminal. Este requerimiento supone uno de los principales retos de los entornos colaborativos, que es el gran número de tecnologías que integran, tanto a nivel hardware como software. Debido a esta heterogeneidad, se ha adoptado la utilización de

comunicaciones basadas en Servicios Web con el fin de facilitar la interacción entre las entidades implementadas con diferentes tecnologías, lenguajes de programación y plataformas hardware. No obstante, el hecho de que los Servicios Web hacen uso del formato XML, repercute negativamente en la sobrecarga de los mensajes y por lo tanto, supone una penalización del rendimiento, en comparación con otros protocolos basados en mensajes binarios, como por ejemplo CORBA (Common Object Request Broker Architecture) [2]. En este sentido, cabe destacar que muchas de las aplicaciones colaborativas de alto nivel proporcionadas por los NGCWEs conllevan el acceso a datos o la interacción entre entidades en tiempo real. Por este motivo, y teniendo en cuenta los limitados recursos de procesamiento y almacenamiento de los dispositivos utilizados, así como la naturaleza inalámbrica de los enlaces de comunicaciones, uno de los principales parámetros de diseño de la solución de seguridad consiste en minimizar la carga impuesta por ésta sobre los sistemas finales.

El resto del trabajo está organizado de la siguiente forma. La sección 2 proporciona una breve revisión crítica de las especificaciones de seguridad existentes para Servicios Web, mientras que la sección 3 define la infraestructura de autenticación y autorización propuesta. Asimismo, en la sección 4 se presenta la integración de la infraestructura de seguridad planteada en un escenario real. Por último, la sección 5 concluye el trabajo.

II. Alternativas para la securización de Servicios Web

La integración de sistemas heterogéneos es una necesidad que ha ido surgiendo junto con el desarrollo de nuevas tecnologías y sistemas de comunicaciones, con el propósito de permitir que diferentes plataformas hardware y software puedan intercambiar información entre sí. Con este fin, muchas de las organizaciones y empresas tecnológicas más importantes decidieron trabajar de forma conjunta para desarrollar un lenguaje común para el intercambio de información, haciendo uso de los estándares existentes en el mercado. A partir de este trabajo surgieron los denominados *Servicios Web* (Web Services).

Los Servicios Web hacen uso de arquitecturas basadas en estándares, como por ejemplo XML, para definir interfaces públicos, y SOAP (Simple Object Access Protocol) para el intercambio de mensajes XML. Entre las aplicaciones típicas de los Servicios Web se encuentran la integración de sistemas dentro de una misma empresa y las transacciones entre diferentes organizaciones a través de Internet. En [3] se proporciona una revisión crítica sobre el uso de los Servicios Web, así como una comparación con tecnologías previas basadas en objetos distribuidos.

Como cualquier otro sistema distribuido, los Servicios Web son objeto de un gran número de ataques de seguridad contra la integridad de los datos, la confidencialidad, la autenticidad, etc. En este sentido, hay que tener en cuenta que la securización de interacciones basadas en Servicios Web puede llevarse a cabo a diferentes niveles. Una posibilidad consiste en securizar los Servicios Web a nivel de transporte, utilizando TLS (Transport Layer Security) [4] o SSL (Secure Socket Layer) [5], mientras que otra posibilidad se basa en implementar los mecanismos de seguridad a nivel de aplicación. Tal y como se muestra en [6] y [7], la securización de Servicios Web a nivel de aplicación provoca un impacto negativo en el rendimiento de los mismos, derivado principalmente de la serialización y des-serialización de los datos XML. No obstante, la implementación de mecanismos de seguridad a nivel de transporte también presenta desventajas, como la complejidad añadida a la hora de gestionar diferentes identidades de usuario y sesiones a nivel de aplicación, así como las limitaciones para atravesar servidores proxy que necesiten examinar los mensajes.

Teniendo en cuenta las características específicas de los entornos colaborativos distribuidos, se ha considerado que es más conveniente securizar las interacciones basadas en Servicios Web de los mismos a nivel de aplicación, por diferentes razones. Primero, esta solución es más flexible, ya que no es necesario firmar o cifrar mensajes completos, sino únicamente las partes deseadas. De esta forma, cada una de las entidades involucradas en la transmisión de un mensaje podrá ver aquellas partes del mensaje dirigidas a ella, mientras que la información confidencial permanece inaccesible. Además, hay que tener en cuenta también que de esta forma los mensajes securizados pueden ser enviados sobre diferentes protocolos de transporte, como pueden ser SMTP, FTP y TCP, sin tener que utilizar necesariamente la versión segura de los mismos.

A. Estándar WS-Security

Con el objetivo de hacer frente a los problemas asociados con la securización de Servicios Web, en 2002 OASIS (Organization for the Advancement of Structured Information Standards) introdujo el estándar WS-Security [8]. Este estándar define los mecanismos necesarios para securizar el intercambio de mensajes SOAP a nivel de aplicación, haciendo uso de unas nuevas estructuras de información conocidas como *tokens de seguridad*. Entre los ejemplos típicos de tokens de seguridad se incluyen certificados X509, nombres de usuario y *tickets* Kerberos codificados en XML. WS-Security se basa en incorporar información acerca de estos procedimientos de autenticación en las cabeceras de los mensajes SOAP, así como de cifrar y/o firmar las partes deseadas de dichos mensajes, bien sea parte de las cabeceras, el cuerpo, o cualquier combinación de ambos. De esta forma, gracias al cifrado parcial o total de los mensajes SOAP, WS-Security proporciona confidencialidad de datos, mientras que mediante el firmado y la inclusión de tokens de seguridad, garantiza la integridad de la información transmitida y la identidad del cliente que originó el mensaje.

A pesar de que la especificación define cómo han de ser introducidos algunos tokens de seguridad en las cabeceras de los mensajes SOAP, no proporciona ningún detalle acerca de los mecanismos que deben usarse para obtener dichos tokens. En concreto, los tokens de seguridad se insertan en los mensajes bajo la etiqueta de cabecera <*wsse: security*>.

Además, aunque WS-Security describe un modelo para proteger los mensajes SOAP, los mecanismos definidos por este estándar sólo proporcionan una forma de securizar mensajes SOAP individuales. Esto puede ser adecuado en escenarios en los que la mayoría de las transacciones consisten en un intercambio de mensajes de tipo petición/respuesta. Sin embargo, en la gran mayoría de las

aplicaciones profesionales, el cliente y el servidor necesitan intercambiar varios pares de mensajes SOAP para llevar a cabo una determinada tarea. En este caso, la implementación del estándar WS-Security de forma independiente para cada par de mensajes resulta muy ineficiente.

Asimismo, para poder mantener el estado de cierto cliente que haya interaccionado previamente con un determinado Servicio Web, son necesarios mecanismos de gestión de sesiones; y en muchos casos incluso es necesario también securizar la integridad de la propia sesión, además de los mensajes individuales. La gestión de sesiones no es un problema nuevo en lo que a la seguridad se refiere. De hecho, la mayoría de los protocolos de seguridad utilizados hoy en día incluyen mecanismos para gestionar la autenticación mutua entre las entidades comunicantes, y además proporcionan mecanismos para la distribución del material criptográfico compartido que se utiliza para securizar el tráfico subsiguiente entre ambas entidades. En el caso de los Servicios Web, se han desarrollado también estándares específicos para cubrir esta necesidad.

B. Estándares WS-Trust y WS-SecureConversation

Los estándares WS-Trust [9] y WS-SecureConversation [10] han sido propuestos como extensiones de la especificación WS-Security, con el fin de definir primitivas de mensajes e interfaces que permitan el establecimiento de contextos de seguridad y la derivación de claves de sesión.

WS-Trust proporciona los mecanismos necesarios para solicitar y emitir tokens de seguridad. Con este propósito, introduce el concepto de una nueva entidad conocida como *Servicio de Tokens de Seguridad* (STS - Security Token Service), la cual se encarga de evaluar las solicitudes y emitir los tokens correspondientes. WS-SecureConversation, por su parte, describe cómo un token de seguridad especifico, denominado *Token de Contexto de Seguridad* (SCT - Security Context Token), puede ser utilizado para securizar sesiones de Servicios Web completas, incluyendo diversos intercambios de mensajes SOAP. El SCT es sólo un elemento utilizado para hacer referencia a un *Contexto de Seguridad* (SC - Security Context) compartido entre un cliente y un servidor dados, del cual se derivan las claves necesarias para proteger las comunicaciones entre estas dos entidades. Por lo tanto, se puede ver cómo el funcionamiento de WS-Trust y WS-SecureConversation está fuertemente ligado.

En la Fig. 2 se muestra el funcionamiento básico de estos dos protocolos. Por razones de simplicidad, se ha considerado que el STS y el servidor de aplicaciones final comparten una misma base de contextos de seguridad. Tal y como se ve, los dos primeros mensajes intercambiados corresponden al protocolo WS-Trust, mientras que a partir del tercer mensaje, la comunicación se securiza haciendo uso de la especificación WS-SecureConversation.

El modelo de seguridad definido por WS-Trust se basa principalmente en el servicio STS, el cual es un Servicio Web responsable de procesar las *Solicitudes de Tokens de Seguridad* (RSTs - Request for Security Tokens) emitidas por los clientes y responder con las *Respuestas de Solicitudes de Tokens de Seguridad* (RSTRs - Request for Security Token Responses) correspondientes. En el mensaje RST, el cliente debe incluir, además de otros datos adicionales, algún tipo de token de identidad acerca de sí mismo, así como

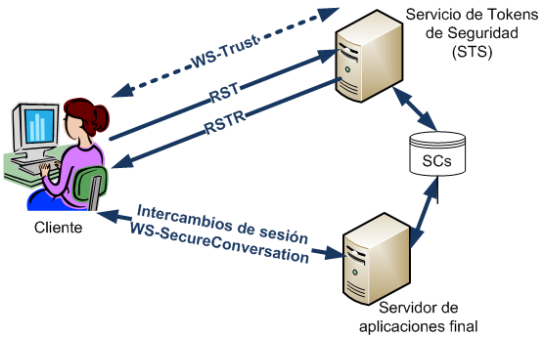


Fig. 2. Intercambio de mensajes típico de los protocolos WS-Trust y WS-SecureConversation

información identificando al servidor destino. Después de validar el RST recibido, el STS genera un nuevo contexto de seguridad para el par cliente/servidor específico, y envía un RSTR al cliente, incluyendo un SCT que hace referencia al contexto de seguridad recién creado. Gracias a la información contenida en el mensaje RSTR, el cliente puede llegar a calcular el mismo contexto de seguridad creado por el STS. En este punto es necesario tener en cuenta que las comunicaciones entre el cliente y el STS están basadas en el intercambio de mensajes SOAP, los cuales a su vez, tienen que ser protegidos. Para ello, puede utilizarse cualquiera de los mecanismos descritos en WS-Security, e incluso podría ocurrir que el cliente necesitara obtener un token de seguridad previo para poder comunicarse con el STS.

No obstante, la especificación de WS-Trust es considerablemente abstracta, ya que proporciona cierta terminología general y una sintaxis XML precisa para el intercambio de datos, pero no define ningún protocolo completo, dejando en manos de las implementaciones concretas algunos aspectos clave, como por ejemplo, el contenido de los contextos de seguridad establecidos. A pesar de que esto puede considerarse como una forma de aumentar la flexibilidad, también pude conducir a confusiones y problemas de interoperabilidad. Por otra parte, WS-Trust no especifica cómo deben llevarse a cabo los controles de autorización. De hecho, esta tarea podría realizarse tanto en el STS como en el servidor final. Sin embargo, los mecanismos de control de acceso constituyen una parte muy importante de una Infraestructura de Autenticación y Autorización (AAI), por lo que deberían estar claramente definidos cuáles son los privilegios asociados con un token de seguridad válido.

Por lo tanto, haciendo uso de la especificación WS-Trust, un cliente y un servidor pueden llegar a compartir un contexto de seguridad común, el cual utilizarán para securizar el subsiguiente tráfico entre ambos, aplicando para ello la especificación WS-SecureConversation. No obstante, hay que tener en cuenta que aunque el caso más común es que WS-SecureConversation se base en la utilización de un SCT, no es obligatorio que sea siempre así, sino que WS-SecureConversation puede utilizar cualquier otro tipo de ticket, como por ejemplo un ticket Kerberos, para derivar las claves de sesión necesarias para cifrar y/o firmar el tráfico.

III. MODELO DE SEGURIDAD PROPUESTO

El objetivo de este trabajo es proporcionar una solución a las necesidades de seguridad de los usuarios de entornos NGCWE, de forma que puedan establecer relaciones de

confianza para el intercambio seguro de datos, garantizando la confidencialidad y la integridad de los datos transmitidos, así como el no-repudio de los mismos. Entre las tecnologías existentes hoy en día para hacer frente a un problema de este tipo, se pueden distinguir dos enfoques básicos: la utilización de infraestructuras de clave pública (PKIs) [11], o los sistemas basados en secretos compartidos, entre los cuales cabe destacar el protocolo Kerberos [12].

La implementación de sistemas basados en PKIs en entornos con unas características tan especiales como las de los NGCWEs presenta grandes retos derivados de la complejidad asociada con la obtención de las claves y los certificados, la verificación de las listas de revocación, el establecimiento de relaciones de confianza entre dominios heterogéneos, etc. En este tipo de entornos, que se caracterizan por la utilización de dispositivos con escasas capacidades de procesamiento y almacenamiento, se consideran más eficientes las tecnologías basadas en secretos compartidos [13]. Por este motivo, en este trabajo se propone un modelo de seguridad basado en la utilización del protocolo de clave simétrica Kerberos.

No obstante, hay que tener en cuenta que tanto el protocolo Kerberos como las soluciones basadas en PKIs son tecnologías de autenticación, por lo que únicamente proporcionan funcionalidades para garantizar la identidad de los clientes, pero no implementan ningún mecanismo para validar los privilegios de dichos clientes, lo que se conoce como proceso de autorización o control de acceso. En consecuencia, son las aplicaciones finales las que han de implementar sus propios mecanismos de control de acceso. Tal y como se ha explicado previamente, en los NGCWEs los elementos básicos se interconectan de forma dinámica para construir aplicaciones colaborativas de alto nivel. En este tipo de sistemas las relaciones de confianza entre los diferentes elementos básicos pueden variar de una aplicación a otra, por lo que no es viable que dichos elementos básicos mantengan y gestionen su propia información de autorización, ya que esto supondría que estas entidades básicas dejaran de ser neutrales e independientes de las aplicaciones colaborativas de nivel superior. Como solución a este problema se propone la implementación de un sistema que gestiona la autorización de forma centralizada, liberando de esta forma a las aplicaciones finales de la necesidad de implementar sus propios mecanismos de control de acceso.

Por otra parte, el hecho de utilizar interfaces basadas en Servicios Web facilita la interoperabilidad entre los diferentes componentes básicos, pero conlleva también una gran penalización del rendimiento, lo cual añadido al hecho de que los elementos básicos están implementados normalmente sobre sistemas hardware de capacidad reducida, puede suponer un problema para la utilización de ciertos dispositivos. Por este motivo, en el modelo de seguridad propuesto, la utilización de comunicaciones basadas en Servicios Web se limita a los casos en los que diferentes sistemas heterogéneos tienen que interaccionar de forma dinámica, es decir, a los casos en los que no se conoce de antemano el extremo de comunicación remoto. En el caso de las interacciones con el servidor de autenticación, sin embargo, teniendo en cuenta que se trata de un servidor conocido, las comunicaciones se basan en el intercambio de mensajes simples de protocolo, sin encapsularlos en mensajes SOAP, limitando así la utilización de esta tecnología computacionalmente costosa.

El modelo de seguridad propuesto consta de dos fases: una fase de autenticación, principalmente basada en el protocolo Kerberos, y una fase de autorización, que supone una extensión de dicho protocolo para proporcionar funcionalidades de autorización.

A. Autenticación basada en Kerberos

Tal y como se ha mencionado previamente, en el modelo de seguridad propuesto la autenticación de los usuarios se lleva a cabo mediante la utilización de Kerberos, el cual es un protocolo ampliamente utilizado para el establecimiento de canales seguros en redes abiertas. En Kerberos, a cada cliente o servidor de un dominio administrativo se le llama *principal*, y cada principal se caracteriza por poseer una clave secreta conocida únicamente por él mismo, y por el *Centro de Distribución de Claves* (KDC – Key Distribution Center) de Kerberos. El sistema de autenticación de Kerberos está basado en la utilización de *tickets*, que no son más que estructuras de información distribuidas por el KDC, que constituyen una prueba de la veracidad de la identidad del principal que los posee. Estos tickets están cifrados de tal forma que únicamente aquellas entidades para las cuales están dirigidos sean capaces de descifrarlos. Por lo tanto, cada cliente que quiera autenticarse frente a un determinado servicio final presentará un ticket emitido por el KDC para dicho servicio. Kerberos incluye mecanismos para evitar la falsificación de la identidad del cliente o del servidor, detectar ataques de repetición, distribuir claves de sesión temporales para el establecimiento de canales seguros, etc. Aquellos lectores que deseen una descripción más detallada del funcionamiento de este protocolo pueden referirse a [14] y [15].

Entre los beneficios que hacen de Kerberos una tecnología adecuada para nuestro enfoque de la solución de seguridad se pueden destacar los siguientes:

- Previene la transmisión de contraseñas en claro a través de la red.
- Proporciona capacidades de Single Sign-On, por lo que cada usuario sólo necesita recordar una única contraseña, e introducirla una vez, para acceder a todos aquellos recursos a los cuales tenga permiso.
- Hace uso de una administración de usuarios centralizada.

Por lo tanto, en la solución de seguridad diseñada, los clientes se autentican frente a un servidor Kerberos haciendo uso del protocolo estándar, de forma que tras la finalización satisfactoria de este proceso, los clientes poseen una clave de sesión, y un ticket de servicio, cifrado con la clave secreta del principal con el que se quieren comunicar, y en el cual está embebida dicha clave de sesión.

Gracias a este material criptográfico es posible enviar al servidor de aplicaciones final una petición de servicio cifrada y autenticada, y securizar después las subsiguientes comunicaciones con dicho servidor. Para probar su identidad, el cliente inserta el ticket de servicio en las cabeceras del mensaje de petición de servicio. Además, cifra también el cuerpo del mensaje con la clave de sesión, garantizando de esta forma la confidencialidad, la integridad y el no-repudio de los datos transmitidos. Cuando el servidor final procesa la petición, primero extrae el ticket de servicio

de las cabeceras de la petición, y haciendo uso de su clave secreta, descifra dicho ticket de servicio obteniendo la información acerca de la identidad del cliente, así como una copia de la clave de sesión utilizada para cifrar el resto del mensaje. Como resultado, el servidor final obtiene la identidad del cliente solicitante de forma segura, ya que ésta estaba cifrada con una clave conocida únicamente por él mismo y por un servidor Kerberos en el que confía. Además, tiene garantías acerca de la integridad y confidencialidad de los datos recibidos, porque la clave utilizada para cifrar esta información sólo la conocen el propio servicio y el cliente legítimo.

A pesar de que el protocolo Kerberos se adapta muy bien a la solución de seguridad propuesta, presenta una gran limitación: únicamente proporciona a los servidores finales fiabilidad sobre la identidad de los clientes, pero no proporciona ninguna información acerca de los privilegios de dichos clientes autenticados. Con el objetivo de hacer frente a esta limitación, se propone el desarrollo de un sistema de autorización centralizado que permite completar las funcionalidades proporcionadas por Kerberos y proporcionar así a los usuarios de entornos NGCWE una solución de seguridad completa e independiente de las aplicaciones de nivel superior.

B. Autorización centralizada

A pesar de que los problemas de la autenticación y autorización en sistemas distribuidos están fuertemente ligados, los esfuerzos invertidos en el desarrollo de mecanismos de autorización seguros para este tipo de entornos no han sido tan significativos como en el caso de los mecanismos de autenticación. En este sentido, cabe mencionar que la mayoría de las infraestructuras de autenticación y autorización existentes actualmente tienden a implementar las funcionalidades de autorización de forma local en los sistemas finales. Sin embargo, en entornos tales como los NGCWEs, no es posible para las entidades finales mantener información de autorización actualizada acerca de todos los posibles usuarios participantes en el entorno colaborativo, ya que a menudo se requiere la interacción entre usuarios y servicios que no se conocen previamente.

En este tipo de entornos es necesario centralizar los procesos de autorización en un único elemento que dé servicio al resto de las entidades participantes, liberando así a los sistemas finales de la necesidad de mantener información de autorización referente a cada uno de los posibles usuarios del entorno colaborativo, así como de la carga de gestión derivada del mantenimiento de dicha información.

Teniendo en cuenta el escenario para el que se ha diseñado la solución de seguridad, compuesto por numerosas entidades que proporcionan funcionalidades básicas y que interaccionan entre sí para crear aplicaciones colaborativas de alto nivel de forma dinámica y bajo demanda, el servicio de autorización centralizado se integra en el sistema como una más de las entidades básicas que lo componen. Por lo tanto, las funcionalidades de autorización son implementadas por un elemento de bajo nivel que se encarga de proporcionar servicios de autorización al resto de las entidades del sistema.

El funcionamiento básico del modelo de seguridad propuesto es el siguiente. Cada vez que un servidor de aplicaciones recibe una petición de servicio, valida la identidad del cliente solicitante gracias al procedimiento de autenticación descrito en el apartado anterior. Una vez que el servidor final ha obtenido la identidad del cliente de manera fiable, pasa a comprobar si dicho usuario tiene o no los privilegios necesarios para acceder al servicio o dato que está solicitando, para lo cual hace uso del procedimiento de autorización descrito a continuación.

El servidor de aplicaciones envía una consulta de autorización al servidor de autorización centralizado, indicando el nombre del cliente solicitante junto con la identidad del servicio solicitado, así como un parámetro que indique el tipo de operación que el usuario quiere realizar (lectura, escritura o ejecución). El mensaje de petición incluye también un sello de tiempo y un campo de información aleatoria que varía en cada intercambio (*nonce*). Tras recibir una petición de este tipo, el servidor de autorización centralizado consulta su base de datos local y envía un mensaje de respuesta al servidor de aplicaciones solicitante, indicando si el usuario está autorizado a realizar la operación solicitada o no. Este mensaje de respuesta incluye entre otros parámetros un sello de tiempo, y el mismo valor *nonce* que contenía el mensaje de petición, para que el cliente pueda reconocer que se trata de la respuesta al mensaje de petición que envió.

Teniendo en cuenta que el servidor de autorización se integra en el sistema como una más de las entidades de bajo nivel que componen el entorno distribuido, la validación de las identidades de los diferentes servidores de aplicaciones que quieran hacer uso de sus servicios se garantiza mediante el procedimiento de autenticación basado en Kerberos descrito previamente, tal y como se muestra en la Fig. 3. Para

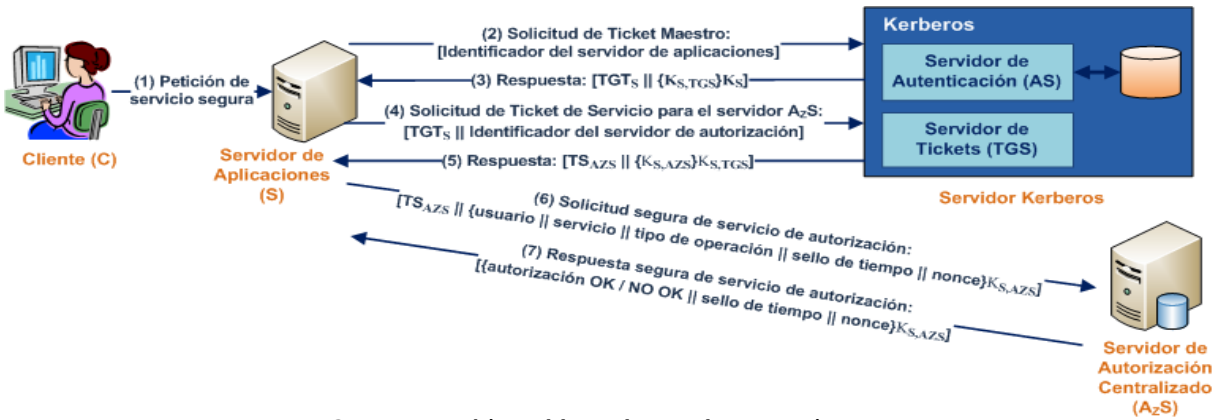


Fig. 3. Interacciones básicas del procedimiento de autorización propuesto

no incurrir en una complejidad excesiva, se ha limitado el contenido de los mensajes representados a aquella información necesaria para entender el funcionamiento de la solución planteada. De acuerdo con la notación empleada, el texto entre corchetes ([ ]) constituye el contenido de los mensajes, mientras que las llaves ({}) delimitan los elementos que viajan cifrados con la clave especificada inmediatamente después.

Según este procedimiento, cada entidad que desee solicitar servicios de autorización tendrá que actuar como cliente del servidor de autorización centralizado, y por lo tanto, obtener del servidor Kerberos una clave de sesión ($K_{S,AZS}$) y un ticket de servicio (TS_{AZS}) emitidos para este servidor de autorización. Haciendo uso de esta información, el cliente construye una petición de servicio de autorización segura, mediante la modificación del mensaje de petición original en dos aspectos: por una parte, inserta el ticket de servicio obtenido (TS_{AZS}) en las cabeceras del mensaje de petición y además, cifra el cuerpo del mensaje con la clave de sesión correspondiente ($K_{S,AZS}$).

Cuando el servidor de autorización centralizado recibe una petición, busca en las cabeceras de dicho mensaje el ticket de servicio correspondiente (TS_{AZS}). Tal y como se ha explicado previamente, este ticket está cifrado con la clave secreta del servidor de autorización centralizado (K_{AZS}), por lo que únicamente esta entidad será capaz de descifrarlo, e incluye la identidad del cliente, así como la clave de sesión necesaria para descifrar el resto del mensaje y proteger las subsiguientes comunicaciones con dicho cliente ($K_{S,AZS}$). De esta forma se evita la posibilidad de que un impostor pueda hacerse pasar por el servidor de autorización centralizado, ya que si la petición de cualquier cliente llegara a un servidor de autorización fraudulento, éste no sería capaz de descifrar el ticket de servicio incluido en la petición (TS_{AZS}) y por lo tanto, no podría llegar a descifrar el resto del mensaje y responder al cliente con un mensaje cifrado con la clave de sesión correcta ($K_{S,AZS}$).

En definitiva, la clave de sesión ($K_{S,AZS}$) que cada cliente envía al servidor final embebida dentro del ticket de servicio sirve para garantizar la identidad de ambos extremos de la comunicación, así como para proteger la integridad y la confidencialidad de los datos transmitidos entre ellos, ya que se utiliza como una clave de cifrado simétrico.

IV. IMPLEMENTACIÓN DEL MODELO DE SEGURIDAD PROPUESTO EN UN ESCENARIO REAL

Tal y como ya se ha mencionado, el trabajo presentado en este documento se ha llevado a cabo en el marco del proyecto integrado C@R "A Collaborative Platform for Working and Living in Rural Areas" del VI Programa Marco. La arquitectura diseñada en este proyecto se basa en un entorno distribuido compuesto por numerosas entidades de bajo nivel que proporcionan funcionalidades básicas y que se comunican entre sí para construir aplicaciones colaborativas más complejas. Todas estas entidades se gestionan desde una estructura centralizada en la que han de registrarse antes de poder interaccionar con el resto. El elemento de gestión centralizado es responsable de coordinar las conexiones entre los elementos básicos, posibilitando así la composición de aplicaciones cooperativas complejas de forma dinámica y en base a módulos pre-escritos.

Por concretar conceptos y a modo de ejemplo, podríamos considerar una aplicación de gestión de situaciones anómalas en un entorno pesquero. Esta aplicación es la que se representa en la Fig. 4. A grandes rasgos, se basa en la utilización de una serie de sensores distribuidos por los barcos que detectan diferentes situaciones anómalas como temperaturas elevadas, humedad excesiva, etc. Las alarmas generadas por estos sensores se clasifican y, en función del tipo al que correspondan, se activan los procedimientos necesarios, como aviso a barcos cercanos, aviso al puerto, establecimiento automático de llamadas de voz o datos, etc. Para permitir el funcionamiento de una aplicación tan compleja, es necesaria la interacción entre diferentes módulos básicos, tales como sistemas de presencia (IMP), servicios de colas de mensajes (MQS), servicios de localización (GPS_LS), etc. La solución de seguridad diseñada permite que las comunicaciones entre todos estos módulos básicos se lleven a cabo de forma confidencial y autenticada.

Dada la heterogeneidad del escenario descrito, ha sido necesario el desarrollo de un módulo software genérico, construido sobre interfaces basadas en Servicios Web, que permita dotar a las entidades participantes en el sistema de las funcionalidades necesarias para poder llevar a cabo las tareas relativas a la solución de seguridad planteada. Este módulo, situado entre las capas de aplicación y de transporte, engloba todas las funcionalidades relativas a los mecanismos de seguridad descritos en el apartado III, permitiendo a los desarrolladores de las entidades básicas abstraerse de los protocolos de seguridad subyacentes.

El módulo software desarrollado está escrito en Java y se ha diseñado como un manejador (*handler*) de Axis2. De esta forma se puede desplegar delante de un servicio y llevar a cabo el procesamiento necesario de los mensajes SOAP. Este modulo constituye una implementación del OASIS Kerberos Token Profile [16], extendiendo la interfaz de programación (*API*) WSS4J [17], de forma que soporte funcionalidades adicionales como la securización de mensajes en base a tickets Kerberos y la consulta del servidor de autorización centralizado.

Sin embargo, debido al hecho de que la utilización de comunicaciones basadas en Servicios Web supone una penalización del rendimiento, y teniendo en cuenta las características específicas del entorno en el que se va a desplegar la solución de seguridad, el cual está

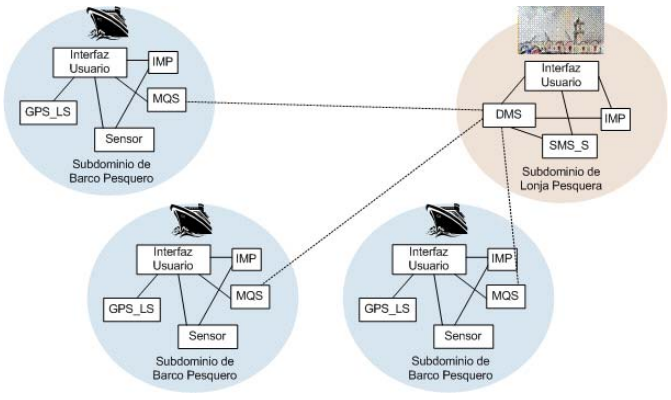


Fig. 4. Ejemplo de aplicación colaborativa de gestión de situaciones anómalas en entorno pesquero

principalmente condicionado por las limitaciones de las entidades que lo componen, se ha decidido restringir la utilización de este tipo de comunicaciones a aquellos casos en los que es realmente necesaria la interacción entre tecnologías heterogéneas de forma dinámica. En los casos en los que las tecnologías a utilizar están previamente fijadas, como es el caso de las interacciones entre cada uno de los clientes y el servidor Kerberos, las comunicaciones se basan en el intercambio de mensajes simples del protocolo, evitando el uso de comunicaciones basadas en Servicios Web computacionalmente costosas. Con el fin de incluir en el módulo software desarrollado las funcionalidades necesarias para interaccionar con el servidor Kerberos se ha hecho uso de la API GSS [18].

A. *Aplicaciones cliente*

En lo que respecta a las aplicaciones cliente, aunque no existe una instancia de Axis2 corriendo, las entidades básicas hacen uso de librerías y funciones Axis2 para negociar el establecimiento de las comunicaciones. En este caso, la parte cliente del módulo software desarrollado intercepta los mensajes SOAP enviados por la capa de aplicación de cada una de las entidades de bajo nivel y procesa estos mensajes llevando a cabo las tareas relacionadas con la securización de los mismos, antes de retransmitirlos hacia el nivel de transporte. En la Fig. 5 se muestra un esquema del módulo desarrollado para las entidades cliente.

Para cada uno de los mensajes SOAP interceptados por el módulo desarrollado, éste identifica al cliente en cuyo nombre se ha enviado el mensaje y el servidor al cual va dirigido. Después de identificar ambos extremos de la comunicación, determina si el cliente posee los tickets Kerberos necesarios y si falta alguno, lleva a cabo automáticamente las interacciones necesarias con el servidor Kerberos para obtenerlo. Principalmente, existen dos razones por las que puede faltar un ticket: porque el cliente no ha necesitado obtenerlo todavía para securizar ninguna comunicación previa, o porque un ticket previamente válido ha expirado.

Una vez que todo el material criptográfico necesario está disponible en la entidad cliente, se cifra el cuerpo del mensaje interceptado utilizando la clave de sesión correspondiente. Con respecto a la inserción del ticket de servicio en las cabeceras del mensaje, el módulo actúa de diferente forma dependiendo de la necesidad de transmitir una nueva clave de sesión al servidor final. Si se trata del primer mensaje que un cliente envía a un determinado servidor, o si el ticket de servicio utilizado para securizar una comunicación ha expirado, el servidor final no puede conocer de antemano la clave de sesión utilizada para cifrar el cuerpo del mensaje, por lo que será necesario que el cliente se la proporcione. Para ello, se inserta en las cabeceras del mensaje el ticket de servicio que contiene dicha clave embebida, de forma que únicamente el servidor final pueda obtenerla, descifrando el mencionado ticket.

A partir de ese momento se utiliza la misma clave de sesión para proteger las comunicaciones entre dicho par cliente/servidor, hasta que expire, lo cual está determinado por el tiempo de vida del ticket de servicio utilizado para transmitirla. Por lo tanto, para las subsiguientes comunicaciones, ambos extremos de la comunicación comparten de antemano un *contexto seguro* con una clave de sesión asociada, lo cual se entiende como el estado al que se llega después de una autenticación y autorización satisfactorias del cliente.

B. *Aplicaciones servidoras*

La parte servidora del módulo software desarrollado tiene que llevar a cabo las operaciones inversas a aquellas realizadas por la parte cliente. Es decir, recibe un mensaje securizado de la capa de transporte y tiene que procesarlo para extraer todas las características relacionadas con la solución de seguridad y, en caso de que el mensaje sea legítimo, proporcionar al nivel de aplicación un mensaje en claro con el formato esperado. La Fig. 6 proporciona un esquema básico del funcionamiento de este módulo.

Cada vez que recibe un nuevo mensaje, la parte servidora del módulo desarrollado inspecciona sus cabeceras para obtener la identidad del cliente que lo envió. Después comprueba si comparte un contexto seguro con dicho cliente, en cuyo caso simplemente hace uso de la clave de sesión compartida para descifrar el resto del mensaje y obtener así una versión en claro del mismo, que reenviará a la capa de aplicación.

Si, por el contrario, el módulo software determina que no comparte un contexto seguro con el cliente solicitante, tendrá que poner en marcha un proceso de validación más largo, ya que no puede confiar en la veracidad de la identidad del cliente ni en sus privilegios de acceso.

Primero tendrá que extraer el ticket de servicio de las cabeceras del mensaje y descifrarlo con su clave secreta, lo que le permitirá obtener de forma fiable información como la

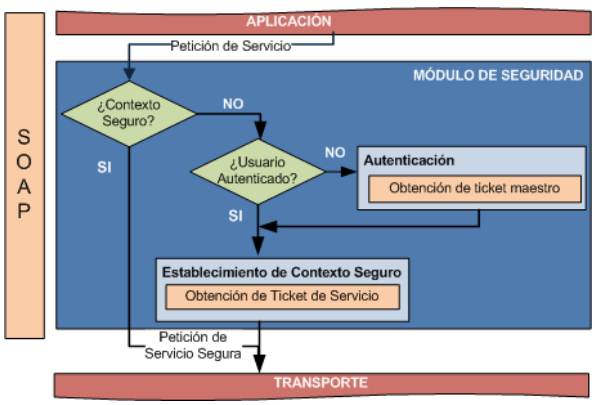


Fig. 5. Parte cliente del módulo de seguridad desarrollado

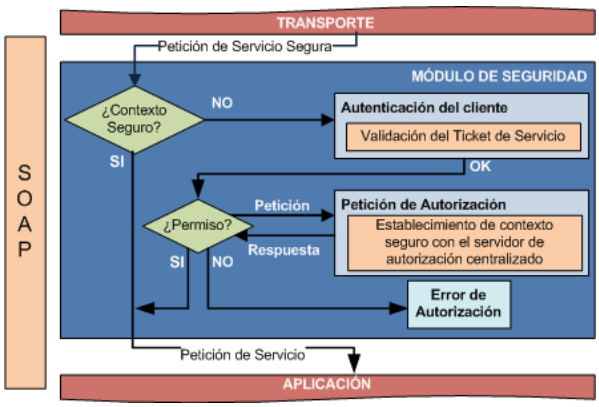


Fig. 6. Parte servidora del módulo de seguridad desarrollado

identidad del cliente solicitante y la clave de sesión necesaria para descifrar el cuerpo del mensaje. Por lo tanto, de esta forma, el servidor de aplicaciones final puede autenticar al cliente, pero todavía no sabe si dicho cliente autenticado cuenta con los permisos necesarios para acceder al servicio o dato solicitado, o no. Teniendo en cuenta que las operaciones criptográficas suponen un alto consumo de recursos, el módulo de seguridad no procederá a descifrar el cuerpo del mensaje hasta que haya comprobado que realmente el cliente solicitante está autorizado a acceder al servicio solicitado.

Con el fin de validar los privilegios del usuario solicitante, el módulo de seguridad emite una petición de servicio de autorización al servidor de autorización centralizado, en la cual incluye, entre otra información, el nombre del cliente solicitante así como su propia identidad. Esta petición de servicio es interceptada por la parte cliente del módulo de seguridad, como cualquier otro mensaje saliente, y protegido correspondientemente. Al recibir la solicitud, el servidor de autorización centralizado consulta su base de información local y responde al servidor final indicándole si el cliente tiene privilegios para acceder al servicio solicitado o no.

Si la respuesta recibida contiene un resultado negativo, el módulo de seguridad de la parte servidora rechazará el mensaje sin descifrarlo, y responderá al cliente con un mensaje de error de autorización. Si por el contrario, la respuesta recibida contiene un resultado positivo, el módulo de seguridad procederá a descifrar el cuerpo del mensaje de petición de servicio, tras lo cual lo reenviará a la capa de aplicación. Por otra parte, también almacenará la clave de sesión asociada al cliente en concreto, en forma de contexto de seguridad, para que en las subsiguientes comunicaciones con el mismo cliente no tenga que llevar a cabo de nuevo los procesos de autenticación y autorización.

V. CONCLUSIONES

Los entornos de trabajo colaborativos distribuidos presentan un gran reto con respecto a la securización de los mismos. Dada su naturaleza distribuida, están expuestos a numerosas amenazas de seguridad, pero debido a sus limitaciones de rendimiento, las soluciones de seguridad utilizadas comúnmente no son adecuadas para estos entornos. En este trabajo se ha llevado a cabo una revisión crítica de las especificaciones de seguridad existentes para Servicios Web y se ha determinado que es posible optimizar su utilización en entornos tan específicos como los presentados. Por ello se ha propuesto una alternativa para securizar dichos entornos de forma eficiente haciendo uso de sistemas criptográficos de clave simétrica y restringiendo la utilización de los computacionalmente costosos Servicios Web a aquellos casos en los que son realmente necesarios.

Por lo tanto, el principal objetivo de la solución de seguridad presentada es liberar a las entidades finales de la necesidad de mantener y gestionar cualquier tipo de información relacionada con los mecanismos de seguridad, e incluso de conocer los protocolos de autenticación y autorización subyacentes. Este hecho supone un gran ahorro de tiempo y esfuerzos para las aplicaciones finales en lo que a las tareas de gestión de usuarios se refiere, ya que tanto la autenticación como la autorización se llevan a cabo de forma centralizada.

Por último, cabe mencionar que el sistema de seguridad propuesto conlleva una ventaja adicional, que es la posibilidad de implementar soluciones de Single Sign-On. Esta característica se deriva de la utilización de tickets Kerberos, ya que una vez que un cliente ha sido autenticado por el servidor de autenticación Kerberos puede obtener todos los tickets de servicio que desee sin necesidad de volver a autenticarse. En lo que al usuario se refiere, la implementación de soluciones de Single Sign-On supone una gran ventaja, ya que no necesita introducir su contraseña constantemente. Además, también implica un aumento de la seguridad del sistema completo, ya que si los usuarios sólo han de recordar una única contraseña es más probable que la protejan y la almacenen de forma segura.

AGRADECIMIENTOS

Parte de este trabajo ha sido financiado por el proyecto integrado C@R “A Collaborative Platform for Working and Living in Rural Areas” (FP6-2004-IST-5 IP) del VI Programa Marco

REFERENCIAS

[1] I. Laso-Ballesteros, “Collaboration@work. At the crossroad of old technology and new IT trends”, *14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise*, pp. 55-65, Jun. 2005.

[2] M. Olson and U. Oqbuji, “Messaging technologies compared”, http://www.ibm.com/developerworks/library/ws-pyth9/

[3] W. Vogels, “Web Services are not distributed objects”, *IEEE International Computing*, vol. 7, no. 6, pp 59-66.

[4] T. Dierks and C. Allen, The TLS Protocol. Version 1.0, IETF RFC 2246, Jan. 1999

[5] A. O. Freier, P. Karlton and P. C. Kocher, The SSL Protocol – Version 3.0, Internet Draft, Transport Layer Security Working Group, Nov. 1996

[6] M. Juric, I. Rozman, B. Brumen, M. Colnaric and M. Hericko, “Comparison of performance of Web Services, WS-Security, RMI and RMI-SSL”, *The Journal of Systems and Software*, vol. 79, pp. 689-700

[7] S. Shirasuna, A. Slominski, L. Fang and D. Gannon, “Performance comparison of security mechanisms for Web Services”, in *Proceedings of 5th IEEE/ACM International Workshop on Grid Computing*, pp. 360-364, 2004

[8] OASIS Web Services Security: SOAP Message Security 1.0 (WS-Security 2004). http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf

[9] OASIS Web Services Trust Language (WS-Trust). Version 1.3. http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html

[10] OASIS Web Services Secure Conversation Language (WS-SecureConversation). Version 1.3. http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/ws-secureconversation-1.3-os.html

[11] R. Housley, W. Polk, W. Ford and D. Solo, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, IETF RFC 3280, Apr. 2002

[12] C. Neuman, S. Hartman and K. Raeburn, The Kerberos Network Authentication Service (V5), IETF RFC 4120, Jul. 2005

[13] M. T. EI-Hadidi, N. H. Hegazi and H. K. Aslan, “Performance analysis of the Kerberos protocol in a distributed environment”, in *Proceedings of the Second IEEE Symposium on Computers and Communications*, pp. 235-239, Jul. 1997.

[14] C. Kaufman, R. Perlman and M. Speciner, *Network security, private communication in a public world*, Prentice Hall Series in Computer Networking and Distributed Systems, pp. 265-325, 1995

[15] W. Stallings, *Network and internetwork security: principles and practice*, Prentice Hall, pp. 315-333, 1995

[16] OASIS Web Services Security Kerberos Token Profile. http://www.oasis-open.org/committees/download.php/16788/wss-v1.1-spec-os-KerberosTokenProfile.pdf

[17] Apache WSS4J. http://ws.apache.org/wss4j/package.html

[18] J. Linn, Generic Security Service Application Program Interface. IETF RFC 1508, Sep. 1993

Selección de Gestores sobre una Arquitectura de Gestión Jerárquica y Distribuída para Redes Personales

José A. Irastorza, Ramón Agüero, Luis Muñoz
Departamento Ingeniería de Comunicaciones,
Universidad de Cantabria
Avd. Los Castros, S/N, 39005 Santander.
e-mail: {angel, ramon, luis}@tlmat.unican.es

***Resumen*- Este artículo recoge el concepto de las redes personales sobre el escenario de las futuras redes de comunicaciones inalámbricas, profundizando en el estudio del marco de gestión para el despliegue de dichas redes. Es ampliamente conocido que las tareas de gestión son fundamentales a la hora de poner en operación cualquier tipo de infraestructura de comunicaciones. Las arquitecturas de gestión usadas en entornos de gestión tradicionales estaban basadas habitualmente en modelos centralizados, los cuales no son aptos ni para las características particulares de las redes personales ni para las topologías multi-salto que las soportan. En este trabajo se propone un modelo jerárquico y distribuido y a su vez se analizan diferentes estrategias para seleccionar de una forma óptima, aquellos nodos que realizan el papel de gestores. Igualmente se ha desarrollado un simulador propietario que permite evaluar las ventajas e inconvenientes de dichos mecanismos, para ello se han estudiado una serie de métricas (probabilidad de que un nodo participe en la arquitectura de gestión, número de saltos necesarios para alcanzar al gestor, y la justa distribución de la carga de gestión entre gestores) Finalmente se ha propuesto una novedosa heurística que perfecciona una de las estrategias analizadas mostrando un rendimiento mejorado sobre el resto de los algoritmos.**

***Palabras Clave*- Redes personales, Modelo de organización de gestión, Modelos distribuidos y jerárquicos, Topología de gestión de red, Algoritmos.**

I. INTRODUCCION

La evolución y uso masivo de dispositivos y periféricos inalámbricos ha sido tan importante como el desarrollo de las tecnologías de red para interconectarlos. Las redes resultantes de este desarrollo están condicionadas por una serie de características particulares a los entornos inalámbricos: movilidad de los nodos, heterogeneidad de los dispositivos y tecnologías de comunicación, así como restricciones en el uso de ancho de banda y energía. Un escenario típico podría incluir varios tipos de dispositivos, desde modernos ordenadores portátiles hasta sensores o actuadores de bajo coste y capacidad, interconectados vía e.g. una red inalámbrica multisalto, la cual puede desplegarse de forma autónoma sobre un área limitada geográficamente, a la cual podemos denominar "Personal Network" (PN). Los desarrollos de estas redes están fundamentados en una conectividad subyacente formada por topologías de redes malladas o multi-salto.

Las topologías multi-salto o malladas son una forma relativamente nueva de desplegar redes de comunicaciones, donde los nodos son dispositivos inalámbricos que, actuando juntos de forma coordinada, crean de forma instantánea y arbitraria una estructura de comunicaciones para compartir e intercambiar información. Una de las características intrínsecas a este tipo de redes proviene del hecho que los nodos se mueven de forma libre y como consecuencia, las conexiones entre ellos son mantenidas mediante el uso de unos protocolos de enrutamiento específicos que son capaces de reconfigurar de forma dinámica la topología de la red dentro de un escenario de comunicaciones multi-salto. Estas redes han reunido el interés de la comunidad científica y se ha puesto especial atención en las llamadas redes ad hoc, en este sentido cabe resaltar el papel de liderazgo investigador que ha tomado el grupo IETF MANET. Sin embargo estos escenarios, que han servido para justificar los desarrollos de estas redes, caracterizados por un número relativamente grande de nodos, que establecen dinámicamente una red de comunicaciones y normalmente sin una infraestructura previa subyacente que lo soporte, no coinciden con los realmente apropiados para una red personal inalámbrica. En estos últimos entornos el objetivo es permitir unas comunicaciones seguras y un acceso a un amplio rango de dispositivos y servicios, incluyendo una conexión con la infraestructura subyacente disponible.

La gestión es tema crucial para cualquier tipo de red y en especial para las redes personales. Por lo tanto, las tareas de gestión asociadas a las redes personales son consideradas como primordiales a la hora de facilitar el funcionamiento efectivo de estas redes. Diferentes arquitecturas y modelos de gestión han sido ampliamente estudiados sobre infraestructuras de redes fijas, no así ha ocurrido con las redes personales, las cuales añaden nuevas dificultades a la tarea de gestión dada sus específicas características. En primer lugar los enlaces de comunicaciones en una red multi-salto inalámbrica son intrínsecamente, poco fiables, dinámicos (debido al movimiento de los nodos) y muestran capacidades variables. Además .los nodos presentan una serie de restricciones, como limitaciones en las baterías y sus capacidades de procesamiento. La principal consecuencia es que la topología resultante no es predecible y por lo tanto necesita de procedimientos automáticos bajo demanda de

reconfiguración. Todos estos hechos imponen una serie de requerimientos a la tarea de gestión y en general a cualquier servicio que vaya a implementarse sobre estas redes. Cuestiones como procedimientos/protocolos de descubrimiento, reconfiguración de los nodos y la topología, seguridad y sobrecarga de la señalización, por nombrar los mas relevantes, deben ser resueltos con el objetivo de gestionar de forma eficiente una red personal.

Por lo tanto, el objetivo es proponer un marco de gestión que considere los citados requerimientos. La elección del modelo de organización será una de las más importantes decisiones de diseño a tener en cuenta. Como se ha dicho anteriormente, las particularidades de las redes personales (y la topología multi-salto subyacente) demandan una diferenciación de los modelos tradicionales usados en las redes fijas. Con todo, las actuales arquitecturas de gestión continúan respondiendo a las dos cuestiones tradicionales: qué se necesita gestionar, y cómo se realizan las tareas correspondientes.

El trabajo presentado en este artículo está centrado en los aspectos relacionados con la cuestión "cómo gestionar redes personales", para lo cual se propone un modelo de organización que distribuye óptimamente el papel de gestor de forma que se consiga gestionar el mayor número de nodos de la red, así como balancear la carga de gestión y minimizar la sobrecarga introducida por el tráfico de red correspondiente.

Hay bastantes trabajos que han analizado las implicaciones de las topologías multi-salto sobre la red de gestión, dos de los más ampliamente referenciados son el marco GUERRILLA [1] y "Ad Hoc Network Management Protocol" [2], planteando ambos una propuesta jerárquica. El primero emplea una división de los nodos en clusters, mientras el segundo utiliza unas sondas activas introduciendo cierto grado de inteligencia en el modelo. El trabajo presentado en [3] es relevante desde el punto de vista que introduce un completo modelo de información e implementa un prototipo de arquitectura de gestión basada en sondas. Es común a los tres anteriores trabajos que analizan tanto el modelo de organización como el modelo de información. Otros trabajos centran su contribución en el modelo de organización, poniendo especial énfasis en arquitecturas de gestión basadas en una combinación de modelos distribuidos y jerárquicos. De entre ellos son destacables [4], [5] y [6], principalmente basados en técnicas específicas de clustering, o [7], que analizan como distribuir las operaciones de gestión de forma óptima en la red.

Para tratar adecuadamente los objetivos propuestos anteriormente, el artículo se estructura de la siguiente forma: La sección II presenta aspectos relevantes sobre el modelo de organización para una arquitectura de gestión a medida de las características específicas de los entornos de redes personales. La sección III plantea como los papeles de agente y gestor se distribuyen entre las entidades de la red, identificando los parámetros que tendrán que tenerse en cuenta para evaluar si la sección es apropiada. A demás se describen diferentes estrategias de selección de los gestores, abarcando desde las más pesimistas hasta las situaciones casi óptimas. La sección IV evalúa estas estrategias mediante el uso de un simulador propietario y se discuten los pros y contras de las mismas. Finalmente, la sección V concluye el artículo con la exposición de algunos aspectos que pueden tratarse en futuros trabajos.

II. ARQUITECTURA DE GESTIÓN DISTRIBUIDA Y JERÁRQUICA PARA REDES PERSONALES

Se puede afirmar actualmente que las futuras redes personales conllevan unos nuevos requerimientos, que incluyen una mayor dinamicidad en lo que respecta a su configuración no estructurada y su topología auto gestionada.

Así, para asegurar una gestión eficaz de las futuras redes personales, formadas usualmente por topologías multi-salto de un número importante de nodos móviles y heterogéneos, se propone utilizar un paradigma de gestión adaptativo y descentralizado. En este sentido, se solventarán los inconvenientes de los sistemas de gestión tradicionales, caracterizados por modelos de organización centralizados y estáticos.

Casi todos los marcos de gestión están basados en el modelo tradicional gestor/agente, que asume un esquema de control centralizado, el cuál es claramente desapropiado para gestionar redes multi-salto o malladas, ya que no es una alternativa razonable, depender de una única entidad central para la gestión de la red y dado que los nodos permanecen intermitentemente conectados. En estos tipos de configuraciones, el gestor representaría un único punto de fallo, algo que no es admisible en las futuras redes personales.

Otra limitación bien conocida del modelo centralizado es su ausencia casi total de escalabilidad. En este sentido, asumir una única estación gestora podría provocar el intercambio de un gran tráfico de gestión entre esta y los agentes correspondientes, así como una alta carga de procesado soportado sobre dicho nodo gestor, lo que podría derivar en un unos altos tiempos de ejecución para las operaciones de gestión. Este hecho incluso se acrecienta sobre las redes personales donde la importancia de las operaciones de gestión (contabilidad, configuración, etc.) es más destacada y por lo tanto la sobrecarga resultante puede alcanzar valores muy altos con el consiguiente empeoramiento del rendimiento de la red. En resumen, se puede decir, que algunas características intrínsecas de las redes personales y las topologías multi-salto/malladas, como la temporalidad de los enlaces, los recursos limitados y el escaso ancho de banda, imponen una metodología para el marco de gestión diferente de la tradicional centralizada.

Con el propósito de superar los inconvenientes, algunos trabajos han propuesto soluciones que refinan el esquema básico centralizado. Una de las más relevantes es la propuesta llamada Management by Delegation [8], la cual fomenta la delegación de ciertas tareas asignadas al gestor hacia los correspondientes agentes haciendo uso scripts descargables de forma dinámica. Otras extensiones de este esquema centralizado están basadas en el establecimiento de cierta jerarquía de agentes, permitiendo una interacción directa entre los agentes. Sin embargo, estas mejoras (todavía

basadas en un modelo centralizado), no solventan eficazmente todos los inconvenientes enumerados anteriormente y por el contrario refuerzan la idea de usar arquitecturas de gestión basadas en modelos descentralizados y dependen de mecanismos de gestión autónoma. Siguiendo esta idea, en este trabajo se presenta un modelo de organización basado en una estructura distribuida y jerárquica.

El marco de gestión que se propone está lógicamente estructurado siguiendo una jerarquía de tres niveles, compuesta por un gestor de nivel superior, el cual puede ser seleccionado de entre un número de gestores de segundo nivel. Estos toman un papel de gestor local, controlando un conjunto de nodos los cuales pueden entenderse como un cluster (caracterizado por algún tipo de conectividad entre sus componentes). Los agentes, entonces, se localizan en el tercer nivel de la jerarquía. Aunque se definen tres niveles, solo existen dos planos de comunicación de gestión: uno conformado por los agentes y su correspondiente gestor (segundo nivel), y otro que interconecta todos los gestores de segundo nivel entre ellos y el gestor global (nivel 1). Como puede verse en la Fig. 1, este plano gestor crea una red superpuesta "overlay" por encima de la red personal que interconecta los gestores de segundo nivel por medio de enlaces lógicos o virtuales, cada uno de los cuales puede corresponder a una ruta particular la cual puede componerse por varios enlaces físicos, bien sobre la red subyacente, bien usando otro servicio de comunicación paralelo.

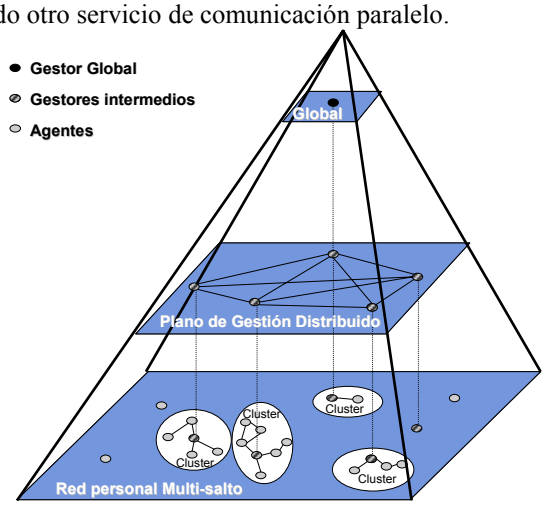


Fig. 1. Modelo de organización distribuido/jerárquico para redes personales

La arquitectura propuesta supone un plano de gestión distribuido (red superpuesta) con un número de nodos que toman el papel de gestor, cada un de ellos controlando una subred (o porción de red) y comunicándose con el resto de los gestores bajo una modalidad colaborativa y entre iguales. Esta propuesta distribuida permite que el subsistema de gestión pueda lograr una mayor fiabilidad y eficiencia, así como una menor sobre carga tanto en las comunicaciones como en los recursos de sistema.

Adicionalmente a este plano distribuido, la arquitectura de gestión también presenta una propuesta jerárquica, ya que el papel de gestor está distribuido entre dos diferentes niveles: el superior representa al gestor global, mientras que los gestores de segundo nivel pueden entenderse como gestores intermedios. Cada uno de ellos controla su propio dominio (porción de la red o cluster), recogiendo y procesando información proveniente de sus respectivos agentes y reenviando en caso de que fuese necesario estos datos al gestor global. También entrega información de gestión desde el gestor global hacia sus propios nodos del dominio. Como se puede observar, el marco de gestión mostrado sigue un modelo de organización distribuido y jerarquizado.

El modelo de organización planteado, presenta una serie de ventajas derivadas de su estructura jerárquica, resaltando una menor sobrecarga del tráfico de señalización de gestión y escalabilidad. Sin embargo, aunque este modelo sea apropiado para los desarrollos de una red personal estándar, no se podrá conseguir una implementación exitosa sino se realiza una buena selección de aquellos nodos que vayan a realizar el papel de gestores. Este problema debe tratarse considerando la topología de la red subyacente, que usualmente conlleva rutas inalámbricas multi-salto, típicas en los futuros desarrollos de las redes personales. De este modo, el marco de gestión necesita establecer los roles de gestor/agente para todos los nodos de la red. Este problema nos presenta algunas cuestiones interesantes que deben ser tratadas, e.g. el número óptimo de gestores que se necesitan desplegar y como se deben situar, el número de agentes que se asignará a cada uno de los gestores desplegados, la probabilidad de que un agente este cubierto por al menos un gestor, y el compromiso entre esta probabilidad de cobertura con el resto de los parámetros como por ejemplo la sobrecarga que supone el tráfico de gestión.

III. PLANTEAMIENTO DEL PROBLEMA

Durante esta sección se asumirá que la red cuenta con N nodos, M de los cuales tomará el papel de gestor, mientras que los A restantes (N-M) son agentes. Además, se asume que la cantidad de agentes cubiertos (esto es, que pueden acceder al menos a un gestor) es AC.

Ya se ha visto en la sección anterior que la mayoría de alternativas plausibles para la arquitectura de gestión a ser empleada sobre una topología multi-salto (como la que se podría requerir en una red personal) promueven repartir la carga de gestión correspondiente entre un número de nodos, resultado, de ese modo, en arquitecturas de gestión jerárquicas/distribuidas. Por tanto, el problema radica en cómo afrontar la selección óptima del conjunto de gestores. Para poder juzgar la idoneidad de la selección, es necesario establecer un conjunto de aspectos "de mérito", cuya combinación podría verse como la que deriva en la estrategia de selección óptima. A continuación se enumeran tres de los más relevantes.

- Probabilidad de ser gestionado. Se trata, posiblemente, del parámetro más obvio, ya que se refiere al porcentaje de nodos que son capaces de comunicarse con, al menos, un gestor y que, por tanto, pertenecen a la arquitectura de gestión. El objetivo sería, por tanto, alcanzar una cobertura total, situación en la que todos los nodos estarían cubiertos al menos por un gestor.
- Número medio de saltos. Uno de los problemas que tradicionalmente se les achaca a las redes multi-salto

es la **interferencia** que las comunicaciones pueden llegar a generar. Para evitar un incremento notable de la sobrecarga asociada a los procesos de gestión, sería deseable requerir un número pequeño de saltos entre cada agente y su gestor correspondiente.

- Distribución de los agentes. La razón principal de fomentar un reparto de la carga asociada a la gestión es evitar la concentración de una cantidad de tráfico en un único punto; en este sentido, la selección de gestores debería buscar un reparto equitativo de los agentes. Para poder caracterizar este aspecto, se define el siguiente parámetro:

$$\beta = \frac{1}{M}\sum_{m}\frac{\left|A_m - \frac{A_C}{M}\right|}{\frac{A_C}{M}} = \frac{1}{A_C}\sum_{m}\left|A_m - \frac{A_C}{M}\right| \qquad (1)$$

Que tiene en cuenta la diferencia relativa global entre la distribución óptima (en la que todos los gestores tienen el mismo número de agentes asociados AC/M) y la distribución actual. Cuanto menor sea el valor de este parámetro, más cerca se estará del reparto óptimo.

Además de los tres parámetros descritos previamente, se analizará asimismo la combinación de los dos primeros, estudiando la probabilidad de que un nodo sea gestionado al establecer un límite en cuanto al número de saltos que es posible emplear para alcanzar a un gestor.

Se analizarán cuatro estrategias diferentes para seleccionar los gestores, tal y como se describe seguidamente.

A. Estrategia 1. Selección aleatoria de gestores

En este caso se asume que los M gestores se seleccionan de manera completamente aleatoria, sin ningún tipo de planificación previa. Esto refleja un escenario poco alentador, ya que, dependiendo de la topología concreta de la red, se podría dar el caso de que hubiera gestores completamente aislados, sin ningún nodo en su área de cobertura. Desde un punto de vista de implementación, esta alternativa no entraña ningún tipo de dificultad.

B. Estrategia 2. Selección óptima de gestores sin conocer la topología

Se asume que los gestores se sitúan en aquellos puntos que aseguran una cobertura (geográfica) máxima de toda el área bajo análisis. Sin embargo, el escenario resultante no garantiza una cobertura máxima de los nodos de la red, pues esto dependerá de su posición concreta en cada topología de red analizada.

C. Estrategia 3. Selección óptima de gestores con conocimiento topológico

Se hace uso de la topología de la red para asignar los gestores de manera óptima. Para ello, se resuelve el tradicional problema de la p-mediana [9], que busca desplegar las 'M' facilidades entre los 'N' nodos de la red. Sin embargo, es bien sabido que uno de las desventajas que principalmente se achacan a esta solución es que busca cubrir todos los nodos, por lo que es posible que existan gestores aislados, lo que podría no reflejar la mejor solución posible.

D. Estrategia 4. Selección sub-óptima de gestores con conocimiento topológico

Como se ha adelantado previamente, una de las desventajas que habitualmente se le atribuye al problema de la p-mediana es que su objetivo primordial es el de cubrir todos los nodos (o lo que es lo mismo, satisfacer toda la demanda). Dependiendo de la topología concreta de la red, podría darse situaciones en las que resultaría más apropiado dejar de gestionar algún nodo, si eso perjudicara a la estrategia de selección de manera global. Para resolver de manera más apropiada este aspecto, se propone una sencilla heurística, según la que no se tendrán en cuenta los subgrafos que tengan un número limitado de nodos, resolviendo el problema de la p-mediana con el resto de la red. En este caso, un parámetro de diseño adicional vendría dado por el tamaño de los subgrafos que se deberían 'cortar' de la red. Es necesario llegar a un compromiso entre la pérdida que suponen los nodos que dejan de gestionarse y el beneficio adicional que se podría llegar a obtener al dejarles fuera de la arquitectura de gestión. Claramente, el borrar los subgrafos de tamaño uno es una alternativa razonable, ya que esos nodos no podrían comunicarse con el resto de la red de cualquier manera. Sin embargo, es necesario analizar si resulta beneficioso cortar subgrafos con un número mayor de nodos.

IV. RESULTADOS

En esta sección se analizan los resultados obtenidos al aplicar los algoritmos de selección de gestores presentados previamente, basándose en una exhaustiva campaña de simulaciones. Se aplicó el tradicional método de Montecarlo, en el que se estudiarán las métricas descritas con anterioridad.

El estudio se lleva a cabo con un simulador propietario, mientras que el problema de la p-media se resuelve gracias a la herramienta popstar [9]. Teniendo en cuenta que el principal objetivo de este trabajo es el análisis de diferentes estrategias de selección de gestores, se asume un modelo de propagación ideal. Hay que decir que la incorporación de modelos más realistas no debería añadir demasiada complejidad al simulador, ya que se podría asumir un alcance medio en dichos casos; además, con este enfoque tan genérico, se podría extender el análisis a otro tipo de escenarios, por ejemplo la gestión de redes de sensores inalámbricos.

Se asumirá un área de 500x500 m2, sobre la que se despliegan, de manera aleatoria, 80 nodos (Proceso de Poisson Point); este valor se ha escogido teniendo en cuenta que es razonable para los escenarios que se están considerando en el trabajo (redes personales). Por otro lado, el alcance de la tecnología inalámbrica empleada se fija a 75 metros. El siguiente paso es el de seleccionar los gestores, atendiendo a los cuatro algoritmos descritos con anterioridad para, posteriormente, analizar la distribución de los agentes entre los gestores seleccionados. Para ello, se estudiará el efecto adicional de incrementar el número de agentes, variando M entre 1 y 10. Cada uno de los casos concretos se simulará 100 veces de manera independiente, asegurando intervalos de confianza adecuados.

Antes de presentar los resultados obtenidos, es necesario establecer el parámetro de diseño que se corresponde con el cuarto algoritmo analizado. La Fig. 2 muestra la función densidad de probabilidad (pdf) del número de subgrafos existentes en el escenario que se empleará para evaluar las diferentes estrategias de selección de gestores y que ya fue descrito anteriormente.

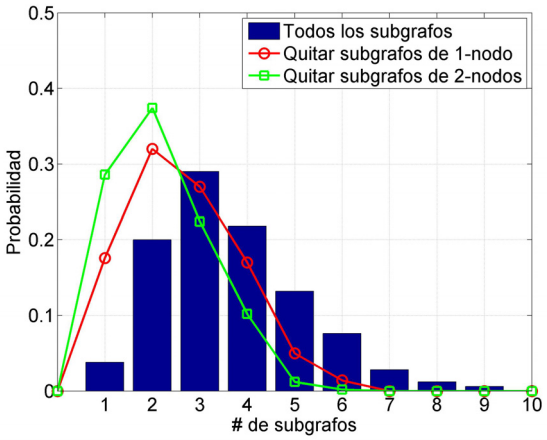


Fig. 2. pdf del número de subgrafos para una red de 80 nodos sobre un área de 500x500 m2. Cobertura fijada a 75 m.

Como se puede ver, cortando los subgrafos de tamaño 1 y 2, el beneficio adicional que se obtiene es relevante, ya que con únicamente 5 gestores se cubriría toda la red en más del 90% de los casos. Además, descartando esos nodos de la red, se asegura que el tamaño de los subgrafos restantes sea mayor, con lo que la solución del problema de la p-mediana debería ofrecer resultados más apropiados, especialmente en lo que se refiere al número de saltos necesarios para alcanzar a un gestor, teniendo en cuenta que habría un número mayor de gestores para cubrir lo que quedase de la red original. Por tanto, se asumirá que un nodo no podrá tomar el papel de gestor (de hecho, no participará en el subsistema de gestión) si el número de nodos con conectividad local (incluyéndose a sí mismo) es menor de 3.

Una vez que la heurística para la novedosa estrategia que se ha propuesto en el trabajo se ha establecido, se puede pasar ya a analizar los resultados obtenidos al emplear cada uno de los algoritmos. La Fig. 3 muestra la probabilidad de que un nodo esté cubierto por, al menos, un gestor. Como puede verse, la estrategia que antes se describió como la peor posible es la que arroja unos resultados más deficientes. Por otro lado, no se observa una diferencia notable entre las estrategias 3 (p-mediana tradicional) y 4 (en la que se eliminan los subgrafos de tamaños 1 y 2), ya que permanece por debajo del 3% para todos los casos. El efecto de eliminar dichos nodos en la red aparece reflejado claramente en la probabilidad de cobertura, ya que ésta tiende asintóticamente a un valor ligeramente a 1 para la cuarta estrategia, mientras que la cobertura total se alcanza prácticamente con sólo 5 gestores en el caso de la p-mediana tradicional. Los resultados también ponen de manifiesto que la estrategia óptima sin conocimiento de la topología no es capaz de alcanzar las prestaciones logradas al ser consciente de cómo es la red en cada caso (p-mediana).

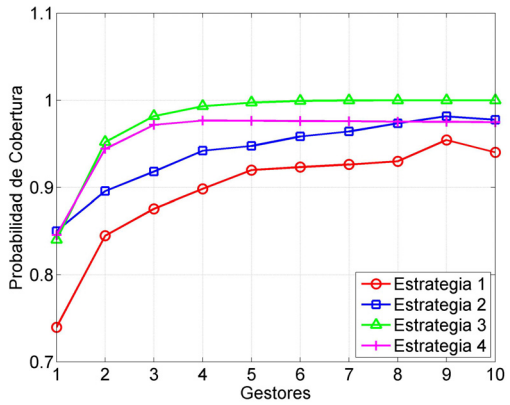


Fig. 3. Probabilidad de cobertura para las cuatro estrategias de desarrollo.

Uno de los beneficios que la heurística propuesta debería aportar sobre la p-mediana tradicional es que debería ofrecer un reparto más equitativo de los agentes entre los gestores disponibles. La Fig. 4 muestra cómo las diferentes estrategias distribuyen la carga de gestión entre los gestores, utilizando el parámetro que se definió previamente (Ec. 1). En este caso, se observa un resultado bastante importante, ya que la tercera estrategia (aplicando la p-mediana sobre toda la red) muestra un comportamiento similar al de la peor de las estrategias (en lo que se refiere a la cobertura, esto es, la estrategia 1), al menos cuando el número de gestores no es elevado. Esto es consecuencia de reservar gestores para cubrir todos los subgrafos de la red, sin importar cuál es su tamaño. Sin embargo, con la modificación propuesta (estrategia 4), la distribución se mejora considerablemente, alcanzando las prestaciones de la mejor alternativa (estrategia 2) cuando el número de gestores es mayor de 4.

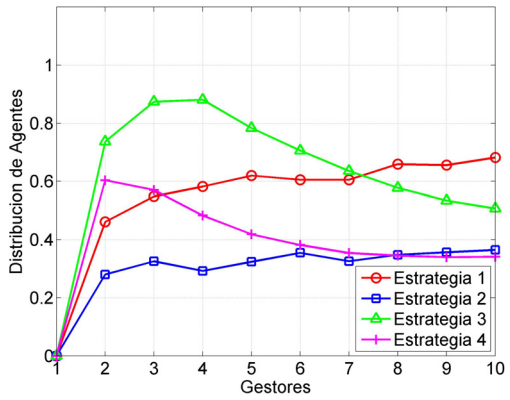


Fig. 4. Distribución de los agentes entre los gestores

El tercero de los aspectos que se identificó como un parámetro a ser optimizado es el número medio de saltos necesarios para alcanzar un gestor. Cuanto menor sea este parámetro, menos sobrecarga generará el tráfico de gestión. Como se puede ver en la Fig. 5, la heurística que se ha propuesto en el trabajo tiene un comportamiento muy similar al de la segunda estrategia e, incluso, es algo mejor para más de 4 gestores. Obviamente el escenario aleatorio es el que peores resultados presenta, mientras que en el caso de la p-mediana tradicional, se observa que, para pocos gestores, el número medio de saltos es elevado, alcanzando prácticamente los resultados de la estrategia aleatoria; esto se debe a que el principal objetivo de la p-mediana es el de cubrir todos los nodos (tal y como se ha dicho anteriormente) y, por tanto,

establece gestores en todos los subgrafos, con la consecuencia de que, en aquellos en los que el número de nodos es elevado, se requieran más saltos para alcanzar el gestor seleccionado, ya que se disponen de menos gestores para cubrir dichos subgrafos.

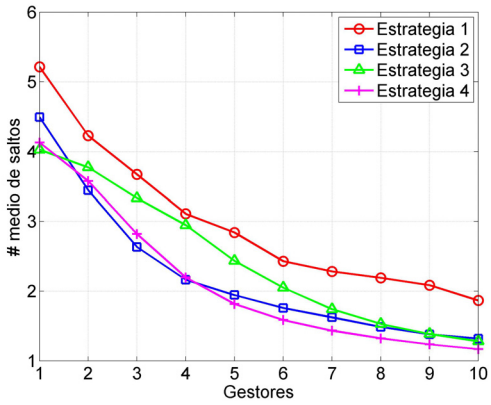


Fig. 5. Número medio de saltos requeridos para alcanzar un gestor

Una conclusión que podría alcanzarse al analizar los resultados de las Figuras 3 y 5 es que, si se estableciera un límite en el máximo número de saltos que podría usarse para alcanzar un gestor, el comportamiento del esquema propuesto podría ser incluso sensiblemente mejor que el del resto de las alternativas analizadas. Como se puede ver en la Fig. 6, que muestra la función de distribución (cdf) de la probabilidad de cobertura cuando se establece un máximo de 2 saltos en las rutas y se despliegan 5 gestores, la solución propuesta es la que ofrece un rendimiento mejor, ya que la probabilidad de estar conectados con rutas de uno o dos saltos es sensiblemente mayor con esta estrategia. En este caso resulta interesante destacar la gran mejora al comparar con los resultados de la p-mediana. Aunque este algoritmo (estrategia 3) alcanzaba una cobertura mayor (ya que no eliminaba ningún nodo de la red), esta ventaja no se percibiría si se estableciera un límite en el número máximo de saltos que podrían utilizarse para alcanzar un gestor.

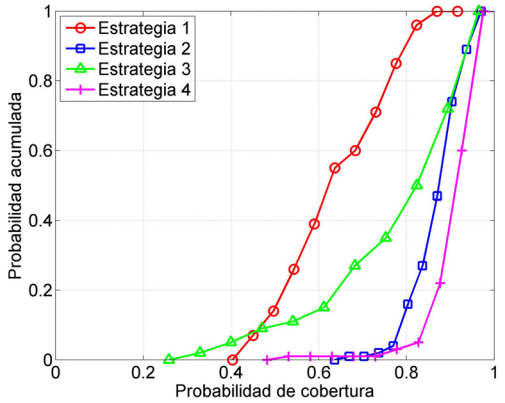


Fig. 6. cdf de la probabilidad de alcanzar, al menos, un gestor cuando el máximo número de salto se limita a 2. Se fija el número gestores desplegados a 5

El resultado anterior pone de manifiesto que la heurística propuesta ofrece unas prestaciones adecuadas para afrontar la selección de gestores en una arquitectura de gestión jerárquica/distribuida, que podría usarse sobre los futuros escenarios de redes personales. A pesar de que se ha utilizado un despliegue de red concreto, resultados obtenidos con diferentes densidades de nodos permiten extrapolar el comportamiento observado a éstas situaciones.

Otro aspecto importante a destacar es que, a pesar de que los análisis llevados a cabo se pueden catalogar como fundamentales, se ha tenido en cuenta en todo momento la posibilidad de afrontar un análisis más centrado en una implementación real. Así, teniendo en cuenta el tamaño de los subgrafos que se proponen eliminar en la estrategia 4, se deriva que la información necesaria podría obtenerse a través de un sistema de monitorización de vecinos, que es razonable pensar esté en cualquier arquitectura de red personal.

V. CONCLUSIONES

Huelga decir que la importancia de los futuros escenarios de comunicaciones inalámbricas seguirá creciendo en un futuro próximo. Uno de los elementos claves de estos entornos son las llamadas redes personales, las cuales están pensadas para proveer a distintos tipos de servicios de un acceso persistente y seguro sobre topologías dinámicas, que probablemente incluyan comunicaciones multi-salto. Tal y como se ha argumentado en multitud de ocasiones, uno de los aspectos fundamentales que garantizan el despliegue de las arquitecturas de comunicaciones o sistemas es dotarlas de un marco de gestión eficiente.

Si consideramos las características intrínsecas de las tecnologías inalámbricas, en general, y las redes personales en particular, no podemos dar por validos los requerimientos de diseño tradicionales que se han venido aplicando a las redes cableadas convencionales. En este sentido el modelo centralizado, tan ampliamente usado y aceptado, en el que un nodo se ocupa de la gestión de toda la red, no se adapta a las necesidades de los nuevos escenarios que se proponen en este trabajo. Como oposición a este modelo, se ha propuesto una arquitectura distribuida y jerárquica, en la cual la caga del gestor está compartida entre un número de nodos, presentando un sistema más escalable y aminorando la sobrecarga asociada al tráfico de gestión. Estos gestores están agrupados sobre una red superpuesta, sobre la que se intercambia la información de gestión. Además, se puede añadir un nivel superior opcional, que puede ser muy útil en casos que un único nodo necesite reunir toda la información de gestión.

La elección apropiada de los papeles de gestor es uno de los temas que con más detalle debe ser abordado en las arquitecturas anteriormente descritas. Este trabajo ha puesto especial interés en tratar este problema. Se han identificado una serie de métricas que deberán analizarse para comprobar cuan apropiada es la selección, contabilizando si se realiza un reparto proporcional de la carga de gestión, la probabilidad de que cualquier nodo participe en el subsistema de gestión y la sobrecarga resultante (basada en el número de saldos que se requieren para comunicarse con el correspondiente gestor). Basándose en estas métricas, se han analizado distintas estrategias: un algoritmo algo pesimista, en el cual los gestores se despliegan aleatoriamente, y dos propuestas más optimistas, una basada en la posición geográfica de los nodos gestores y otra en la topología particular de la red. Para esta última, se ha propuesto una mejora basada en una nueva

heurística, consistente en la eliminación de aquellos componentes de red (subgrafos) que tienen un número relativamente pequeño de nodos.

De los resultados se derivan dos conclusiones principales: por una parte, es fundamental llevar a cabo una selección apropiada de los gestores, ya que pueden haber grandes diferencias según la estrategia particular de selección; además, la heurística propuesta presenta unos resultados muy interesantes, ya que aporta un mayor rendimiento en términos de la distribución de los agentes entre los gestores seleccionados, al tiempo que tiene en cuenta el número de saltos empleados para alcanzar el gestor correspondiente, mientras no se penaliza sustancialmente la probabilidad de cobertura.

El estudio que se ha realizado en este trabajo se ha basado en teoría de grafos, pero una visión mas amplia ha suscitado la posibilidad de mapear los algoritmos propuestos sobre protocolos y desarrollos de redes reales. En es sentido, y gracias al marco de simulación de gestión que fue presentado en [10], nos permite, como trabajo de futuro, seguir analizando las implicaciones de la arquitectura de gestión propuesta así como de las arquitecturas de selección de gestor en bases al rendimiento de las comunicaciones y protocolos.

AGRADECIMIENTOS

Los autores desean expresar su agradecimiento al Proyecto Nacional de I+D del Ministerio de Educación y Ciencia titulado "Optimización de Técnicas de Descubrimiento de Servicios sobre Plataformas Inalámbricas Heterogéneas" (TEC2006-05819).

REFERENCIAS

[1] C-C. Shen, C. Srisathapornphat, and C. Jaikaeo, “An Adaptive Management Architecture for Ad Hoc Networks”, IEEE Communications Magazine, vol. 41, no. 2, February 2003, pp. 108-115

[2] W. Chen, N. Jain, and S. Singh, “ANMP: Ad Hoc Network Management Protocol” IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, August 1999, pp. 1506-1531

[3] R, Badonnel., R, State, and O, Festor, “Management of Mobile Ad Hoc Networks: information model and probe-based architecture”, International Journal of Network Management, vol. 15, Issue 5, September 2005, pp. 335-347.

[4] S. Sivavakeesar, G. Pavlou, and A. Liotta, "Stable Clustering Through Mobility Prediction for Large-Scale Multihop Intelligent Ad Hoc Networks", Proc. of WCNC 2004, Atlanta, USA, March 2004.

[5] L. Fallon, D. Parker, M. Zach, and M. Leitner, Sandra Collins: “Self-forming Network Management Topologies in the Madeira Management System”, Proc. of AIMS 2007, Oslo, Norway, June 2007, pp. 61-72.

[6] R. Badonnel, R. State, and O. Festor, “A Probabilistic Approach for Managing Mobile Ad Hoc Networks”, Transactions on Network and Service Management, vol. 4, no. 1, June 2007, pp. 39-50.

[7] K-S. Lim, C. Adam, and R. Stadler, “Decentralizing Network Management”, KTH Technical Report, December 2005.

[8] Y. Yemini, G. Goldszmidt and S. Yemini, “Network Management by Delegation”, Second International Symposium on Integrated Network Management IM’91, Washington, D.C., April 1991, pp. 95-107.

[9] M. G. C. Resende and R. F. Werneck, “A hybrid heuristic for the p-median problem” Journal of Heuristics, vol. 10, Issue 1, January 2004, pp. 59-88.

[10] Jose A. Irastorza, R. Agüero and L. Muñoz, “Fostering the simulation-based evaluation of management architectures over multi-hop topologies“ IEEE/IFIP Network Operation and Management Symposium (NOMS’08), Salvador do Bahia, Brazil, April 2008.

Desarrollo de un agente SNMP v3 para modelado de usuario en entornos LAN

N. Lasierra, M. Lopez, J. García, A. Alesanco
Grupo de Tecnologías de la Información (GTC) del Instituto en Investigación en Ingeniería de Aragón (I3A)
Universidad de Zaragoza
50018, Zaragoza
nelia.lasierra@unizar.es, marcs.lopz@gmail.com, jogarmo@unizar.es, alesanco@unizar.es

***Resumen*—En este artículo se presenta el desarrollo de un agente SNMP v3 (*Simple Network Management Protocol*) para el modelado de usuarios en entornos LAN (*Local Area Network*). Este agente establece comunicaciones SNMP tanto con los gestores de red encargados de configurar el proceso del modelado, como con los usuarios de los que recoge información contenida en las MIB (*Management Information Base*) con el objetivo de encontrar un patrón que caracterice su comportamiento. Esta información será procesada y analizada por una red neuronal tipo SOM (*Self Organizing Map*), que permitirá, tras el proceso de aprendizaje, la detección de anomalías respecto al comportamiento normal del usuario. Tanto los parámetros a configurar para definir el modelado de cada usuario como los resultados de la supervisión del agente quedan recogidos en la MIB del modelado contenida en el agente propuesto. De esta forma, el agente desarrollado facilita una herramienta única para modelar a todos los usuarios de la misma red LAN y constituye un sistema totalmente integrado en la arquitectura SNMP. Por último, se presenta un escenario de pruebas para la aplicación de la detección de intrusos del agente propuesto. Se ofrece la información seleccionada para modelar al usuario, así como parámetros de configuración y primeros resultados obtenidos.**

***Palabras Clave*—MIB, modelado de usuarios, SNMP, SOM**

I. Introducción

Conocer las preferencias, costumbres, utilización de recursos o de forma general, información que identifique o caracterice el comportamiento de un usuario, es actualmente un ámbito de gran interés, impulsado por el crecimiento de las redes y la expansión de Internet. Extraer esta información, que permite adecuar el diseño de páginas web al perfil de los usuarios o la personalización de búsquedas guiadas entre otras aplicaciones, es una necesidad que se extiende a redes de área local, donde conocer el comportamiento de sus usuarios proporciona información de gran utilidad para, por ejemplo, distribuir y ofrecer de forma diferenciada el acceso a los recursos disponibles en la red [1]. Especialmente, esta técnica es de gran utilidad en redes en las que existen limitaciones de ancho de banda o existe la necesidad de aplicar técnicas de QoS (*Quality of Service*). Este proceso de extracción y evaluación de la información que permite determinar el comportamiento de un usuario se conoce como modelado del usuario. Sin embargo, la distribucción de recursos no es la única aplicación del modelado en redes LAN. Puesto que la seguridad es uno de los problemas de mayor preocupación que se plantea hoy en día en el mundo de la informática y las redes de comunicaciones, la aplicación más popular del modelado de usuario en redes LAN es la detección de intrusos. A diferencia de otras soluciones tecnológicas desarrolladas para este fin, como la detección de patrones o firmas de ataques conocidos (fundamentalmente existen estas dos técnicas para la detección de intrusos), la idea que implica este proceso es, una vez identificado el comportamiento habitual de un usuario, detectar posibles anomalías en el mismo que se deban a la presencia de un intruso. Esta técnica abre una nueva vía para la detección de intrusos que trata de solventar las limitaciones encontradas en los sistemas anteriores, como por ejemplo, la detección de nuevos atacantes hasta el momento desconocidos por el sistema. Esta tarea del modelado de usuario, implica por tanto una intensa recogida de datos que debe hacerse de forma transparente al mismo y para lo que se hace necesario la búsqueda de técnicas de gestión de redes que permitan este intercambio de información de forma sencilla. Una solución atractiva para esta tarea es la utilización de la arquitectura SNMP [2]. Esta arquitectura de gestión proporciona un interfaz estándar que facilita las comunicaciones entre diferentes dispositivos de red y pone a disposición del gestor gran cantidad de información de forma estructurada a través de las MIB tanto de parámetros de tráfico como de la utilización de recursos o información de configuración del dispositivo

Si bien es cierto que la utilización de SNMP para el modelado de tráfico de usuarios es una apuesta ampliamente extendida [3] y algunos estudios proponen integrar el desarrollo de un detector de intrusos en esta arquitectura [4], hasta lo que nosotros sabemos, ninguna solución propuesta para el modelado de usuarios ofrece una solución global en términos de escalabilidad, seguridad, facilidad de comunicaciones, procedimiento de identificación de patrones y generalización de aplicaciones. Por ello, en este artículo se presenta el desarrollo de un agente SNMP v3 como solución global para el modelado de usuario. Este agente, que aprovecha las ventajas de esta arquitectura tanto para las comunicaciones como para el acceso y almacenamiento de la información analizada, va a permitir que uno o varios gestores integrados dentro de una arquitectura de gestión de red SNMP puedan modelar el comportamiento de todos los usuarios situados en la misma red de área local así como su continua supervisión y detección de posibles anomalías en su comportamiento habitual gracias al desarrollo de una red SOM que analizará la información extraída de cada usuario.

II. Modelado de Usuario

Podemos definir el modelado de usuario como un proceso a través del cual obtenemos una representación de las características relevantes de su comportamiento. De forma general,

este proceso se compone de tres fases: 1) selección de información 2) caracterización y aprendizaje del comportamiento y 3) supervisión y detección de anomalías.

A. Selección de información relevante

La primera tarea a desarrollar para encontrar un patrón que caracterice el comportamiento de un usuario es seleccionar e identificar la información que queremos extraer del mismo. La elección de estos parámetros vendrá condicionada por la aplicación concreta para la que se quiera conocer su comportamiento. Por ejemplo, en experiencias como [1], en las que el objetivo del modelado es clasificar a los usuarios en diferentes categorías o perfiles para la posterior distribución de recursos en el acceso a Internet, se propone la utilización de parámetros de tráfico HTTP, SMTP y FTP. Entre otros, se analiza el ancho de banda utilizado, la duración de las conexiones, tamaño de los paquetes enviados y los puertos utilizados con el objetivo de identificar el uso de aplicaciones *peer to peer* en concreto. Sin embargo, si la aplicación es la detección de intrusos, otros estudios como [5] y [6] proponen para modelar al usuario utilizar parámetros de tráfico ICMP, TCP y UDP (número de paquetes enviados y recibidos, paquetes erróneos, análisis de cabeceras o número de conexiones, etc) así como otros parámetros internos como el uso de memoria y CPU o los comandos tecleados por el usuario. El compromiso que plantea mantener la eficiencia de red y obtener la cantidad de información necesaria para el modelado hace aumentar la importancia de este proceso de selección de parámetros.

B. Caracterización del comportamiento

Una vez seleccionada la información con la que se va a modelar al usuario comienza la fase de adquisición de datos y aprendizaje del comportamiento. Para la recogida de esta información, se ha seleccionado la utilización de la arquitectura de gestión SNMP. Esta es una solución muy popular utilizada por sistemas de modelado de usuario dada la flexibilidad y facilidad de uso que esta arquitectura ofrece, permitiendo de una forma estructurada y efectiva el intercambio de información. Por ejemplo, en [3] y [7] se analiza información contenida en las MIB 2 para la detección de ataques de DoS (Denial of Service). Otros métodos o técnicas ampliamente adoptados para la adquisición de datos de usuario son la utilización de archivos de logs del sistema operativo o sistemas de filtrado de tráfico [8].

Durante esta fase de caracterización del comportamiento, la información extraída del usuario es procesada y clasificada como normal o habitual para el usuario modelado. La duración de este periodo de aprendizaje debe permitir observar y definir de forma representativa los distintos patrones que puede presentar el usuario. Para esta tarea de caracterización del comportamiento de los usuarios existen múltiples propuestas. Algunas de las técnicas más populares tanto para el modelado de usuarios como de tráfico de los mismos, apoyan su funcionamiento en la utilización de redes neuronales, modelos de markov, estudios estadísticos o lógica borrosa , entre otros [6] [9] [10]. En la solución propuesta, el elemento utilizado para el aprendizaje del comportamiento del usuario y decisión de las posibles anomalías observadas, es una red SOM.

Por último, en la fase de test o supervisión, se establecen periodos de encuesta para recoger información del usuario y comparar el comportamiento observado con el comportamiento clasificado como normal con el objetivo de detectar así anomalías en el comportamiento monitorizado.

III. Arquitectura del Agente SNMP para el modelado de usuario

La arquitectura general del agente SNMP v3 propuesto para el modelado de usuario está compuesto por tres bloques: 1) el bloque de comunicaciones, 2) el bloque de la MIB del modelado, y 3) el bloque de las redes SOM (una por cada usuario a monitorizar). Como se muestra en la figura 1, este agente establece comunicaciones SNMP tanto con los usuarios a modelar como con los gestores que configuran y supervisan el modelado de cada uno de ellos. A través de la MIB del modelado (contenida en el agente desarrollado) el gestor puede configurar los parámetros involucrados en el modelado de cada usuario (individualizando así el proceso para cada uno de ellos) así como acceder a los resultados obtenidos de la supervisión del agente. Estos resultados se consiguen tras el procesado de la información extraída del usuario y posterior análisis mediante una red neuronal SOM. El agente desarrollado constituye una herramienta única para modelar a todos los usuarios de la misma red que evita la utilización de nuevas herramientas software en los equipos a modelar y posibilita su uso en los equipos que ya posean una gestión de redes vía SNMP.

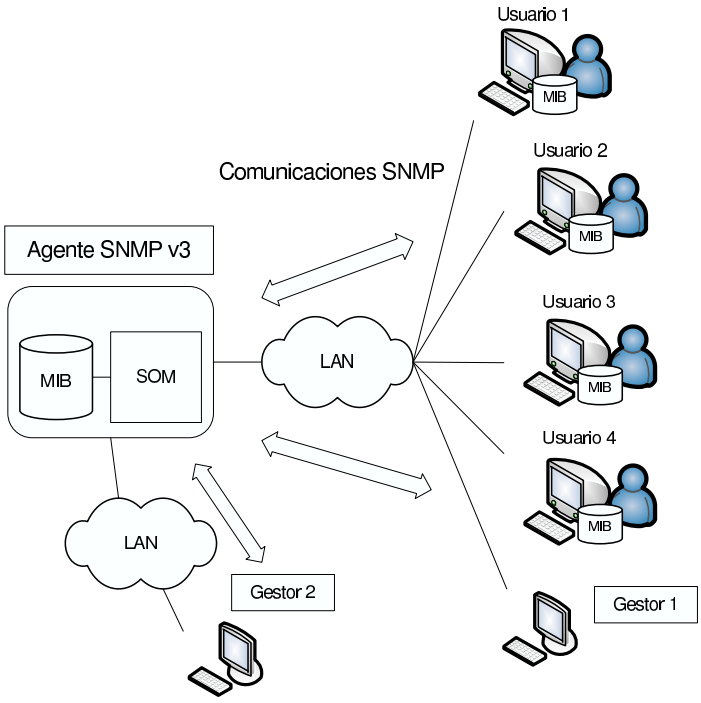


Fig. 1. Funcionamiento general del Sistema.

En la figura 2 se muestra un esquema del funcionamiento e interacción del agente con el gestor y los usuarios, así como de la relación entre los bloques del mismo (comunicaciones, MIB y SOM) que intervienen en cada una de las fases del proceso de modelado del usuario. Mediante el envío de paquetes tipo SnmpGet, el agente extraerá información

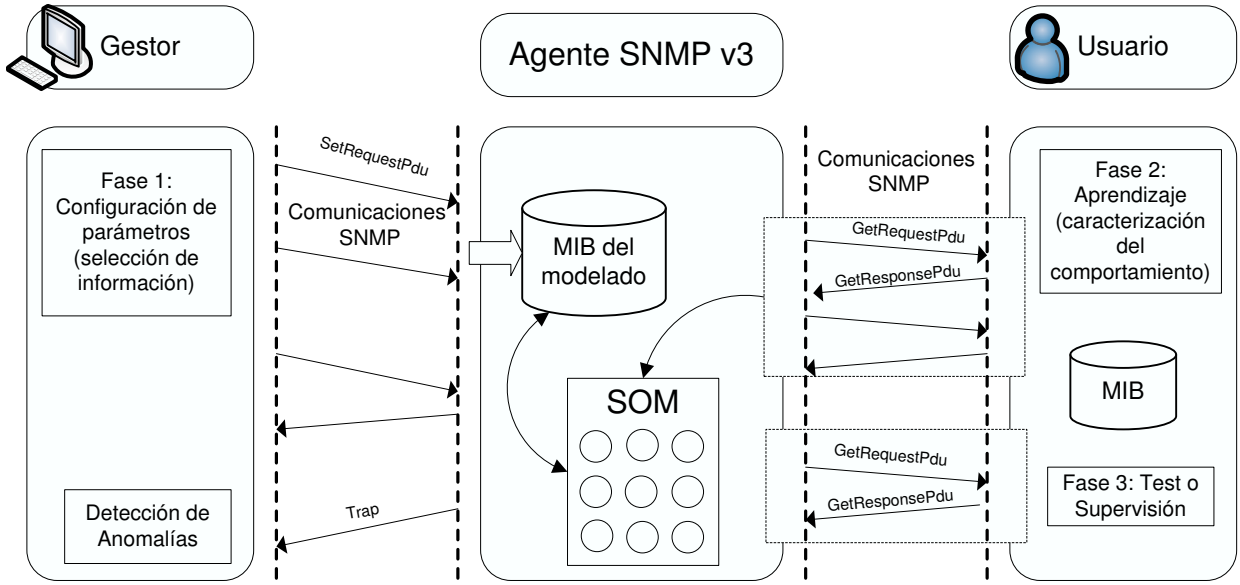


Fig. 2. Comunicacion entre bloques del Sistema.

contenida en las MIB de los usuarios. Esta información será analizada por la SOM asociada al mismo y los resultados obtenidos, almacenados en la MIB del modelado. De esta forma el gestor, a través del envío de paquetes tipo SnmpSet (para la modificación de datos) y tipo SnmpGet (para la lectura de resultados) dispondrá de control total sobre la definición y supervisión del comportamiento de los usuarios. Para el desarrollo de los módulos que componen el sistema se ha escogido el lenguaje de programación JAVA, así como el motor de base de datos MySQL como soporte para la implementación de la MIB propuesta.

A. Bloque 1: Comunicaciones SNMP

Las comunicaciones entre el agente y los participantes involucrados en el proceso del modelado (gestores y usuarios) se realizan mediante el protocolo SNMP v3. Esta versión del protocolo incorpora aspectos de seguridad con lo que el agente diseñado dispone de funciones de autenticación, cifrado y control de acceso [11]. De esta forma queda garantizada que tanto las funciones de gestión como el acceso a la información queda limitado a personas autorizadas. Aunque es recomendable utilizar esta versión, el agente desarrollado soporta además las anteriores versiones del protocolo.

Como se observa en la figura 2, el agente desarrollado se comunica tanto con los gestores que definen y configuran los parámetros del modelado a través de la MIB que implementa, como con los usuarios de los que recoge información para caracterizar y posteriormente supervisar su comportamiento. De esta forma, el agente incorpora dos funcionalidades: por una parte actúa como un agente normal de la arquitectura SNMP capaz de responder a órdenes y peticiones de una estación gestora así como enviar de forma asíncrona información importante no solicitada (traps) y por otra, actúa como un gestor ya que establece comunicaciones SNMP con los agentes de los usuarios para pedir información de forma periódica (lanzamiento de encuestas) acerca de variables contenidas en las MIB (está información puede estar localizada tanto en las MIB2 como en las MIB privadas que implemente el dispositivo).

B. Bloque 2: MIB UserModelling

La MIB del modelado, contenida en el agente desarrollado, constituye una estructura de almacenamiento de todos aquellos valores que intervienen en este proceso. Su implementación ofrece flexibilidad en la definición de parámetros (lo que permite individualizar y particularizar el modelado para cada uno de los usuarios de la red de acuerdo a los criterios del gestor) y permite disponer de un módulo de fácil acceso para el gestor en el que se exponen de forma estructurada los resultados de la monitorización del usuario. Esta MIB que ha sido denominada *UserModelling* y está situada dentro de las MIBs privadas (puesto que el esquema propuesto de modelado sólo se aplica a terminales personales), aporta escalabilidad al diseño propuesto, ya que sirve como base para futuras aplicaciones que quieran extender las funcionalidades del agente.

La MIB implementada está compuesta por 4 tablas: *userModelingControlTable*, *modelingParametersTable*, *userModelingDataTable* y *logUserModelingTable*. Las tablas *userModelingControl* y *modelingParameters* permiten al gestor configurar todas las variables que intervienen en el modelado del usuario. Estas tablas funcionan de forma lógica como una única tabla de ”Control”, a pesar de estar separadas en su defición en dos para facilitar el proceso de configuración de parámetros, cuyo funcionamiento sigue las reglas de las tablas de control RMON [12] . La tabla *userModelingData* es una tabla de lectura para el gestor, en la que se muestran para varias observaciones realizadas por el agente, los resultados obtenidos del modelado del usuario, lo que permite al gestor un seguimiento temporal de la evolución del comportamiento del mismo. Por último, la tabla *logUserModeling*, recoge información relativa a observaciones del comportamiento del usuario identificadas como comportamiento anómalo. A continuación se describen

con más detalle los objetos contenidos en estas tablas así como las funcionalidades de los mismos:

- ***userModelingControlTable***: La tabla de Control es una tabla de lectura/escritura, en la que se definen: los parámetros para identificar a cada uno de los usuarios a modelar (dirección IP del usuario a modelar e índice del modelado, ambos necesarios ya que cada usuario puede ser modelado por varios gestores y con distintos parámetros), aspectos relacionados con la configuración de la red neuronal (dimensiones, factor de aprendizaje, tiempo de entrenamiento o radio de vecindad entre otros), información relativa para la detección de anomalías (umbrales y eventos asociados), información acerca del gestor que controla cada proceso de modelado configurado así como los intervalos de tiempo involucrados en los periodos de encuesta para el aprendizaje y supervisión del comportamiento. Como se observa en la figura 3, para la recogida de datos del usuario, se realizan encuestas al usuario de forma periódica. La duración de estas encuestas, el tiempo de activación entre ellas, así como el tiempo entre envío de paquetes de petición de datos por parte del usuario al agente, permitirán la configuración de los periodos de test y aprendizaje del comportamiento.

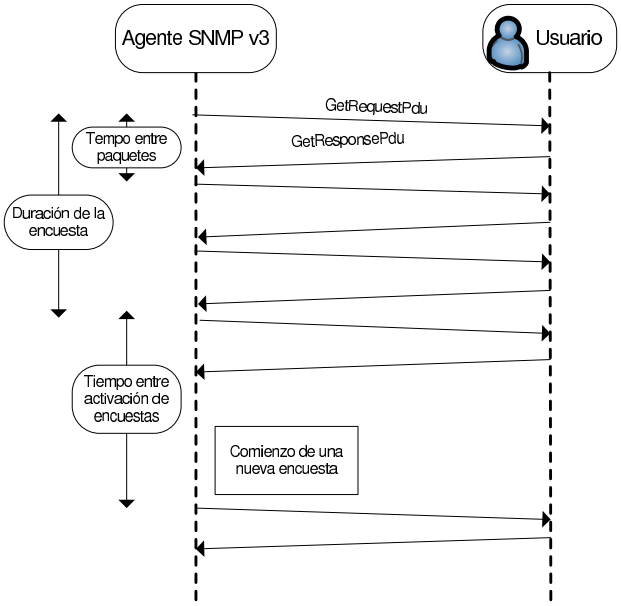


Fig. 3. Esquema de encuestas

- ***modelingParametersTable***: Esta tabla permite al gestor especificar la información contenida en las MIB que va a servir para modelar el comportamiento del usuario. Puesto que muchas veces la información que el gestor considera más interesante para caracterizar el comportamiento del usuario no se encuentra de forma directa en las MIB, sino que es consecuencia de la combinación de dos que si lo están, por cada variable a utilizar en el modelado, el gestor deberá especificar dos OIDs, así como la relación que existe entre ambas para conseguir el resultado deseado.
- ***userModelingData*** : En esta tabla se recogen los resultados de las encuestas realizadas a cada usuario. Es decir, en ella se muestran los valores observados para cada una de las variables definidas, el instante temporal de la observación proporcionada así como la salida de la red neuronal para dicho comportamiento observado. El contenido de esta tabla va a permitir al gestor observar el comportamiento del usuario durante un intervalo de tiempo, ya que asociado a cada uno de ellos se define un buffer circular que permite mostrar la evolución de los valores observados.
- ***logUserModelingTable*** : En función del evento definido en la tabla de control se registrará o no una entrada en esta tabla cada vez que la red neuronal detecte un comportamiento anómalo en el usuario. En esta tabla se muestra la información relativa a todas las observaciones realizadas al usuario que indican claramente un comportamiento distinto al habitual. Su funcionamiento es similar a la anterior, ya que para cada usuario modelado se define un buffer en el que se muestran todas las anomalías detectadas en el periodo de monitorización.

Para notificar al gestor cambios en la evaluación del comportamiento del usuario e informarle acerca de anomalías detectadas, la MIB *UserModelling* define el envío de dos tipos distintos de Traps: *BehaviourAlarm* y *WarningAlarm*. La definición y posterior envío de este tipo de paquetes permite al gestor aumentar los intervalos de encuesta (al agente propuesto) para revisar el comportamiento de los usuarios modelados. De esta forma, se reduce el tráfico de la red y se permite al gestor el conocimiento total sobre cambios en esos parámetros.

C. Bloque 3: SOM.

Para analizar la información extraída del usuario y posteriormente facilitar la toma de decisiones (comportamiento normal o anómalo), se utilizan redes SOM [13]. Los mapas autoorganizados (SOM) son un modelo de red neuronal muy utilizado en la práctica para el análisis y visualización de grandes cantidades de datos. Este tipo de redes se basan en el aprendizaje no supervisado, de forma que la red es entrenada con un conjunto de patrones de entrada sin asociar una salida deseada para cada una de ellas durante esta fase. En función de las similitudes observadas entre los datos de entrada se van formando, de forma adaptativa, grupos de neuronas a la salida con características similares, de forma que se representa en las agrupaciones de las neuronas de la red una estimación de la función de densidad de probabilidad de las entradas. Así, el funcionamiento de estas redes se basa en establecer una correspondencia entre los patrones de entrada y un espacio bidimensional de salida, de forma que los representantes de cada agrupación obtenida queden espacialmente correlados y alejados de aquellos respecto a los que presentan mayores diferencias.

A pesar de que existe una clara tendencia a la elección de redes MLP (*Multilayer Perceptron*) en la mayoría de sistemas en los que se implementa una red neuronal para la toma de decisiones, para el sistema de modelado propuesto la implementación de una red SOM es la más adecuada dado que este tipo de redes presentan propiedades como facilidad de implementación, flexibilidad y capacidad de generalización,

pero fundamentalmente su elección queda determinada por dos razones: 1) el número de comportamientos "normales" que puede presentar el usuario es en principio desconocido y 2) sólo es posible entrenar la red con patrones de comportamiento normal del usuario y no anómalos (ya que son desconocidos). Antes de comenzar el modelado de un usuario, no podemos asumir que el comportamiento observado sea estable en el tiempo, por ello, la estructura utilizada para caracterizar su comportamiento debe permitir identificar cada uno de estos comportamientos observados durante la fase de entrenamiento y aprenderlos como normales para el mismo. Por otra parte, puesto que para la fase de aprendizaje sólo conocemos la información relativa a lo que es normal o habitual en el usuario, el proceso de entrenamiento de la red debe realizarse únicamente a partir de patrones positivos. Dadas las condiciones del sistema diseñado para el modelado, la implementación de una red SOM es sin duda, para esta aplicación, la solución más adecuada.

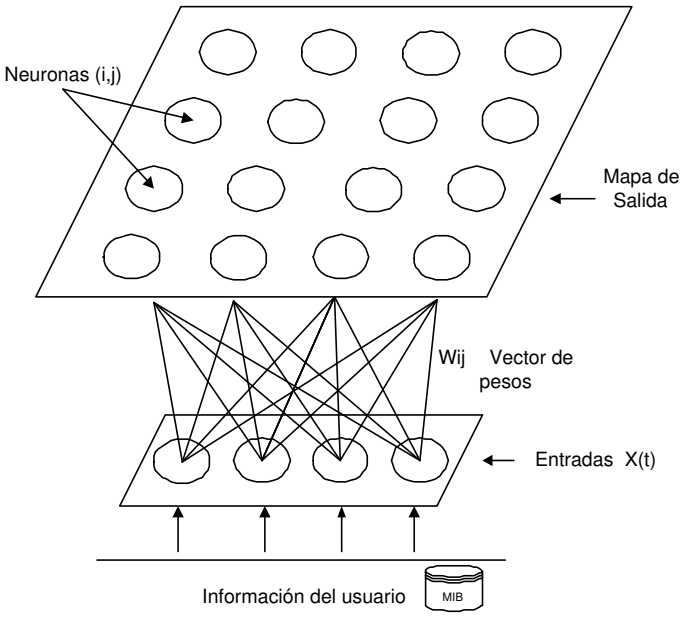


Fig. 4. Arquitectura general de una red SOM

En la figura 4 se muestra la arquitectura típica de una red SOM. La información extraída del usuario será procesada y presentada como entrada a la red neuronal, por lo que el número de elementos del vector de entrada (X(t)) queda determinado por el número de variables seleccionadas para modelar al usuario. El número de neuronas del mapa de salida, queda determinado por las dimensiones especificadas por el gestor. Asociado a cada una de ellas (denominadas en la figura como neuronas ij de acuerdo a su posición en el mapa), existe un vector de pesos wij(t) compuesto por un número de elementos igual al número de variables de entrada, de forma que su situación en el mapa y actualización del vector de pesos está directamente relacionada con las observaciones presentadas a la entrada de la red.

Para cada uno de los usuarios a modelar se asocia una red de este tipo, de forma que la implementación de este módulo va a permitir de forma sencilla, representar todos los posibles comportamientos normales que pueda presentar un usuario. Notar que tanto el número de comportamientos normales que puede presentar un usuario como el perfil de estos comportamientos es diferente de los que pueda presentar otro, ya que lo que para un usuario es normal para otro puede no serlo, por ello es necesario disponer de una red SOM para supervisar de forma individual el comportamiento de cada uno de ellos. Durante la fase se aprendizaje se procesará la información extraída de las encuestas a los usuarios y presentará a la red como datos de entrada normalizados a valores comprendidos en el rango [0,1]. Así, de forma adaptativa se irán actualizando los pesos de las neuronas (inicializados de forma aleatoria) hasta quedar organizada de acuerdo a las observaciones presentadas. Para la actualización de los pesos se ha programado una función gaussiana, ya que dadas sus características, ofrece rapidez de convergencia (lo que asegura una ordenación más rápida del mapa). Tanto para el factor de aprendizaje como para el radio de vecindad de la función gaussiana (parámetros configurables por el gestor) se ha programado una función exponencial, de forma que el ajuste entre la entrada y el vector de pesos de la neurona va disminuyendo a medida que se desarrolla el proceso de aprendizaje así como el radio de vecindad va tendiendo a valores más pequeños para poder incurrir en un error de desajuste mínimo al final del aprendizaje.

Durante la fase de supervisión o test, se presentará el patrón de observación a la SOM (obtenido de la información del usuario) y se indicará la clasificación del comportamiento observado. El agente implementado identifica en base a los resultados de la SOM los posibles comportamientos normales que presenta el usuario (diferenciación de zonas en el mapa de salida de la red) y notifica al gestor tras el procesado de la información extraida del usuario, si la observación del comportamiento es normal (se corresponde con alguna de las zonas detectadas como comportamiento habitual) o es anómala (no se corresponde con ninguno de los comportamientos obtenidos en la fase de aprendizaje). Esta decisión se toma en base a la combinación de dos criterios de distancia: 1) el vector de pesos de la neurona ganadora y 2) la distancia en posiciones a la neurona seleccionada como representativa del comportamiento normal del usuario.

IV. Aplicación a la detección de intrusos

Una de las aplicaciones más populares del modelado de usuarios es la detección de intrusos. Su funcionamiento se basa en encontrar anomalías o desviaciones respecto al comportamiento habitual del usuario que se deban a la presencia de un atacante. Actualmente, nuestro trabajo se centra en el estudio del modelado de usuarios para esta aplicación. A continuación se exponen los estudios realizados para la selección de información utilizada en el modelado del usuario, así como los parámetros seleccionados para la configuración del mismo, escenario de pruebas propuesto y los primeros resultados obtenidos.

A. Selección de Parámetros.

Para realizar una adecuada selección de parámetros se ha llevado a cabo estudio acerca de como la actividad de los diferentes tipos de ataques que existen afectan a la modificación del comportamiento de los usuarios atacados. Podemos

clasificar los diferentes ataques en 4 grandes grupos: ataques de denegación de servicio (DoS), Root Attacks, Remote to User Attacks y Probing [14].

- *DoS*: El objetivo final de estos ataques es la inutilización del sistema víctima. Un elevado consumo de CPU, la saturación de la memoria o la sobrecarga de tráfico IP pueden ser las consecuencias de la actividad de este tipo de ataques.
- *Root Attacks*: Intentan conseguir los privilegios de superusuario en el sistema víctima para robar información valiosa, borrar las huellas del ataque, atacar otros sistemas, etc. La detección inmediata debería ser más complicada, ya que no existen parámetros estándar SNMP para monitorizar los inicios de sesión de usuario en un sistema operativo. Sin embargo, a partir de las acciones posteriores del atacante y a su uso anómalo de los privilegios de usuario se podrá realizar una detección implícita o indirecta del ataque.
- *Remote to User*: Se asemejan al anterior grupo en que el éxito reside en la elevación de privilegios para el atacante. No obstante, el control del sistema es parcial, ya que depende de los privilegios del usuario comprometido en el sistema. El estudio de la posible detección es similar al anterior grupo.
- *Probing*: Consiste en obtener la mayor información posible del sistema atacado. Suele ser precedente de un ataque de otro grupo, es decir, con el probing se estudian las debilidades y el ataque en sí es el uso indebido de ellas. Al igual que ocurre para ataques *DoS*, el estudio de variaciones en parámetros de tráfico (paquetes UDP, ICMP o TCP enviados y recibidos, número de conexiones realizadas, paquetes erróneos, etc), permitirá una detección más sencilla de este tipo de ataques.

Cabe destacar que normalmente, un ataque no pertenece a un sólo grupo ya que está formado por una sucesión de ataques de distinto tipo. Por ejemplo, un atacante podría obtener información del sistema mediante probing, luego obtener privilegios de root usando esa información, para posteriormente utilizar la máquina víctima y realizar así un ataque DoS a un segundo sistema.

En total se han seleccionado 11 variables con las que modelar el comportamiento del usuario. Estos parámetros escogidos, extraídos de la MIB2 (estadísticas de red) y MIB25 (*Host-resources*), son los siguientes:

- (1) Datagramas UDP con puerto desconocido
- (2)(3) Ancho de banda de entrada y salida
- (4)(5) Segmentos TCP de entrada y salida
- (6)(7) Datagramas UDP de entrada y salida
- (8)(9) Número de conexiones TCP y UDP
- (10) Consumo de CPU
- (11) Uso de memoria RAM

La utilización de las 7 primeros parámetros (variables de tráfico) resulta de gran utilidad para la detección de ataques de *Probing*, así como la utilización de los 10 últimos parámetros especificados para el modelado del usuario (todos salvo el primero) proporciona información útil la detección de ataques de *DoS*. De forma general, cualquiera de estos parámetros podría quedar modificado como consecuencia de ataques de *Root* o *Remote to User*, ya que normalmente estos conllevan el lanzamiento de un ataque de DoS o Probing (esto dependerá de las acciones realizadas al adquirir los permisos del sistema).

B. Entorno de pruebas y primeros resultados

Para verificar la eficacia del sistema propuesto, hemos planteado un proceso de pruebas compuesto por dos fases: 1) simulaciones de comportamiento de usuario en entornos virtuales y 2) modelado de usuarios sobre escenarios reales. El objetivo de la primera fase de pruebas ha sido identificar los valores de configuración que garantizan el correcto funcionamiento de la SOM, así como determinar la capacidad de la red neuronal para identificar durante el aprendizaje distintos perfiles de comportamiento normal de un usuario y clasificar posteriormente durante el periodo de test las observaciones realizadas. Para ello, se ha definido un fichero compuesto por 2400 vectores de datos (utilizado para el aprendizaje) que simulan el comportamiento de un usuario cuyo perfil hemos determinado a través de observaciones realizadas a un usuario real con un software de gestión de redes. Para el periodo de test se ha definido igualmente un fichero de comportamiento compuesto por 1500 vectores de test que simulan tanto el comportamiento normal del usuario como comportamientos alterados por la presencia de intrusos. Estos últimos vectores de test se han conseguido a través de modificaciones del patrón original utilizado para el aprendizaje de la red de acuerdo a la forma de actuar de distintos ataques. En concreto se han simulado ataques de DoS que modifican el consumo de la CPU y el volumen de tráfico de entrada y salida, actuando tanto de forma individual como de forma combinada entre ellos.

El número de variables y funciones que se deben elegir en cuanto a la estructura y configuración de la red neuronal, es extenso. Para una convergencia óptima de una red SOM se recomiendan de 50 a 100 iteraciones por neurona aproximadamente [13], por lo tanto, determinar este parámetro es importante ya que constituye una referencia para definir el resto de variables. En base a las pruebas realizadas, el número de iteraciones por neurona óptimo para un correcto funcionamiento de la red se ha estimado en 40. Tras la realización de pruebas con diferentes configuraciones de las dimensiones de la red (número de filas y columnas), se optó por una red 6x4 ya que el mapa de salida de la red permite una representación clara de las zonas de actividad del usuario siendo el tiempo estimado para el aprendizaje del comportamiento(consecuencia de las variables definidas) de 16 horas. Este periodo permitirá identificar totalmente el comportamiento del usuario estudiando su actividad durante dos jornadas laborales típicas, en las que probablemente, el patrón observado se repita.

En la tabla I se recogen los resultados de las pruebas realizadas en la primera fase utilizando la configuración de la red descrita en las líneas anteriores. En ella se detalla el número de comportamientos normales identificados para el usuario así como la tasa de falsos positivos (vectores de comportamiento normal clasificados como comportamiento

anómalo), el porcentaje comportamientos anómalos detectados debido a la presencia de un intruso (Inrusos Detectados) y el porcentaje de comportamientos anómalos no detectados (Intrusos No Detectados). Tras las pruebas realizadas, hemos fijado en 0.28 el umbral utilizado para la detección anomalías (para la distancia euclídea entre vectores de pesos) estableciendo así un compromiso entre el porcentaje de falsos positivos y la tasa de intrusiones no detectadas.

Tabla I
RESULTADOS OBTENIDOS DE LA FASE 1 DE PRUEBAS.

Nº Comportamientos Normales	% Falsos Positivos	% Intrusos Detectados	% Intrusos No Detectados
4	0.9%	86.6 %	13.4%

Actualmente estamos trabajando en segunda fase del proceso de pruebas modelando el comportamiento de diferentes usuarios reales. En las primeras pruebas realizadas, no se ha comprometido el sistema con agentes externos (como puedan ser infecciones víricas o ataques desde otros equipos), aunque el trabajo futuro incluirá la introducción de ataques con la finalidad de evaluar la capacidad del sistema para la detección de anomalías sobre diferentes escenarios. A partir de los resultados obtenidos en la fase anterior, hemos modelado a cada uno de los usuarios durante un periodo de aprendizaje de 16 horas no consecutivas (equivalente a 2 jornadas laborales típicas), eligiendo para ello una red SOM de 11 parámetros de entrada y un mapa de salida de dimensiones 4x6. Posteriormente, hemos supervisado durante 19 horas el comportamiento de los usuarios con el objetivo de determinar la tasa de falsos positivos obtenida con la red propuesta. Para esta fase se han establecido periodos de encuesta de 6 minutos, en las que el tiempo entre envío de paquetes ha quedado fijado a 1 minuto y el tiempo entre activación de encuestas en de 30 minutos. Es importante destacar, que tanto la configuración propuesta de la red como la elección de parámetros de tiempo, permiten de forma general caracterizar el comportamiento del usuario, y que por tanto, la solución propuesta puede ser utilizada en otra aplicación del modelado de usuarios distinta a la detección de intrusos.

Tras la fase de aprendizaje o entrenamiento del comportamiento del usuario 1, hemos diferenciado, en base a los resultados de la SOM, 5 zonas en su mapa de salida. Las zonas 1, 2 y 3 (marcadas en la figura 5) se corresponden con 3 comportamientos normales identificados en el usuario. La zona 4 o zona de transición ha quedado definida por su cercanía a las zonas de comportamiento normal del usuario y por tanto la decisión de comportamiento anómalo tras la activación de una neurona de esta zona en la fase de test vendrá condicionada fundamentalmente por la distancia de su vector de pesos. Por último, la última zona identificada (marcada en la figura 5 con neuronas en blanco) es una zona de neuronas que no han obtenido ninguna victoria (o un número insignificante) durante la fase de aprendizaje por lo que sus pesos no han sido apenas modificados respecto a su inicialización. Por lo tanto, la activación de una neurona de esta zona durante la fase de test, indicará la detección de un comportamiento anómalo en el usuario. Estas zonas quedan representadas en la SOM de la figura 5, en ella, el aumento de la frecuencia de victorias de cada neurona se representa con un color más oscuro en la escala de grises.

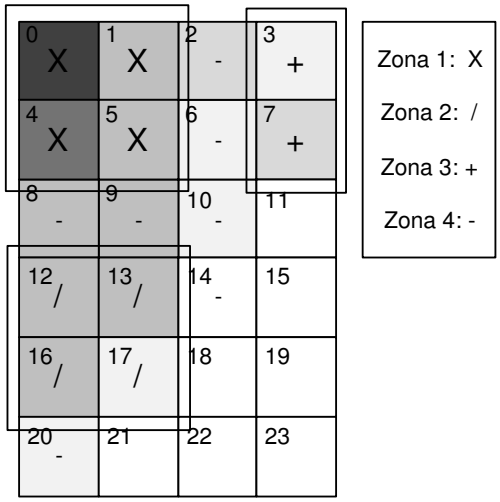


Fig. 5. SOM tras el aprendizaje.

A continuación se resumen los parámetros característicos de cada zona:

- Zona 1: No existe un consumo de ancho de banda apreciable, ni número elevado de conexiones UDP o TCP. El consumo de CPU es muy bajo, en torno al 1%.
- Zona 2: Es una modificación del perfil de la zona 1, en el cual se han establecido más conexiones TCP. El consumo de CPU es ligeramente mayor, aunque sigue siendo bajo, situándose entre el 4 y el 8%.
- Zona 3: Es la zona de mayor uso de procesador, entre el 20 y el 30%, además de una subida del uso de ancho de banda, tanto de subida como de bajada (aproximadamente 1 Mbps en ambos sentidos).
- Zona de transición o zona 4: Son neuronas cercanas a una o más de las zonas anteriores. Sus pesos son intermedios entre los de las zonas adyacentes a ellas o próximos a los de una, si sólo se encuentran cercanas a ella. Han llegado a este estado de pesos como consecuencia de estar situadas en el área de vecindad durante las victorias de las zonas normales, incluso llegando a obtener alguna victoria esporádica en transiciones de un perfil a otro.

Tabla II
RESULTADOS DE LA FASE DE TEST.

Zonas	1	2	3	Transición	Anómala
Porcentaje en test	74,7%	3,2%	5,3%	15,8%	0,53%

Durante el periodo de test, en el se han introducido a la red un total de 190 vectores de entrada correspondientes a observaciones del comportamiento del usuario, se obtienen los datos reflejados en la tabla II. En resumen, un 99,47% de las entradas son catalogadas como normales frente a un

0,53%, que nos indicarían falsos positivos en la detección de comportamiento anómalo.

En base a estos resultados, se concluye que todavía es necesario trabajar en la mejora del procesado previo de los parámetros de entrada para optimizar el modelado de usuario ya que la influencia de variaciones en ciertos parámetros es superior a otros. Sin embargo, los resultados obtenidos en el aprendizaje y test de un comportamiento normal, que presentan una separación de diferentes perfiles en zonas, una sencilla caracterización de cada perfil con los pesos de las neuronas y, por último, una tasa baja de falsos positivos, parecen mostrar una buena aproximación del comportamiento de la red a lo que de ella se espera para nuestra aplicación.

V. Conclusiones y Líneas Futuras

En este artículo se ha presentado el desarrollo de un agente SNMP v3 orientado al control y supervisión de comportamientos de usuarios en entornos LAN. Para ello se han diseñado e implementado los bloques que integran el sistema completo: establecimiento y control de comunicaciones SNMP entre los participantes, redes SOM y diseño de la MIB del modelado. De esta forma, el agente propuesto permite la integración del modelado de usuario en un sistema de gestión de redes global, dejando a elección del gestor la configuración de parámetros necesarios para el proceso en función de la aplicación del mismo. Aunque existe libertad en la elección de estos parámetros, se ha ofrecido una propuesta tanto para la configuración de la red neuronal como para el establecimiento de los tiempos implicados en los periodos de encuesta, que de forma general, pueden ser utilizados para cualquier aplicación del modelado de usuario. Como ejemplo de aplicación, se ha presentado con mayor detalle la utilización del agente para la detección de intrusos. Tras un estudio de las consecuencias que la actividad de un intruso provoca en el comportamiento del usuario, hemos realizado una selección de parámetros, optimizado la configuración del SOM para el aprendizaje del comportamiento del usuario y realizado pruebas muy preliminares sobre un escenario real.

Actualmente estamos trabajando en la optimización del proceso de modelado para la aplicación de la detección de intrusos. Entre las tareas a desarrollar, se encuentra la mejora del sistema para la detección de comportamientos y definición de umbrales, así como el estudio de la correspondencia temporal entre el comportamiento observado y la activación de los comportamientos aprendidos del usuario. Además, se está llevando a cabo la segunda fase del proceso de pruebas. En esta fase se está modelando a usuarios reales con distintos comportamientos y utilizando ataques reales para el periodo de test. Este proceso permitirá generalizar la eficiencia del modelado y selección de parámetros de configuración.

Referencias

[1] K.Mizanian, R.Zakeri, S.Tasharrofi, M.Analoui "User Modeling Using Network Layer Information", *Proceedings of the International Conference on Networking,International Conference on Sytems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06)*,IEEE Publishing, 2006.

[2] J. Case, M. Fedor, M. Schoffstall, J.Davin "Simple Network Management Protocol", *Informe Técnico*, Internet Engineering Task Force(IETF), RFC 1157,1990

[3] J.Li, C.Manikopoulos, "Early Statistical Anomaly Intrusion Detection of DOS Attacks Using MIB Traffic Parameters", *Proceedings of the 2003 IEEE*.

[4] P.Astithas, G.Koutepas, A.Moralis, B.Maglaris "SIDS - A System for enterprise-wide Intrusion Detection", *International System Security Engineering Association Conference '01*, Orlando,USA, 2001

[5] S. Kompella, S. Singh, G. Varghese, "On Scalable Attack Detection in the Network", *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, 2007.

[6] L. Vokorokos, A. Balaz, M. Chovanec, "Intrusion Detection System using Self Organizing Map", Acta Electrotechnica et Informatica, 2006

[7] W. Lee, R. K. Prasanth, B. Ravichandran, R. K. Mehra, "Proactive Detection of Distributed Denial of Service Attacks using MIB Traffic Variables, A Feasibility Study", *Proceedings of the 7th IFIP/IEEE International Symposium on Integrate Network Management*, Seattle, WA, 2001.

[8] M. Barla, M. Bielikova, "Estimation of User Characteristics using Rule-based Analysis of User Logs", *11th International Conference on User Modeling*, Greece,2007.

[9] S. Hyun, W. Suk Lee, "An Anomaly Intrusion Detection Method by Clustering Normal User Behaviour", *Computers and Security*, vol. 22, no. 7, pp. 596-612, 2003.

[10] F. Astithas, G.Koutepas, A.Moralis, B.Maglaris, "Modelling Network Traffic as alfa -Stable Stochastic Processes. An Approach Towards Anomaly Detection", *VII Jornadas de Ingeniería Telemática,JITEL 2008*,2008

[11] J. Case, R. Mundy, D. Partain, B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", *The Internet Society*, RFC 2570, 1999

[12] J. Waldbusser, "Remote Network Monitoring Management Information Base", *Informe Técnico*, Internet Engineering Task Force(IETF), RFC 1757, 1999

[13] B. Martinez del Brio, A. Sanz Molina, "Redes Neuronales y Sistemas Borrosos", 3rd ed,Ed RA-MA ISBN 8478977430, 2006

[14] K. Kendall, "A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems", Master Thesis of Engineering in Electrical Engineering and Computer Science,1999

Arquitectura y diseño de un modelo de red OBS para simulación

Felix Espina, Javier Armendariz, Mikel Izal, Daniel Morató, Eduardo Magaña
Universidad Pública de Navarra, Campus de Arrosadía s/n, E-31006 Pamplona
e-mail: {felix.espina, javier.armendariz, mikel.izal, daniel.morato, eduardo.magana}@unavarra.es

***Resumen*—Optical Burst Switching (OBS) es una nueva tecnología de conmutación óptica capaz de soportar una gran demanda de ancho de banda en backbones ópticos con Wavelength Division Multiplexing (WDM). Muchos investigadores están interesados en el estudio de esta propuesta emergente y la búsqueda de sus parámetros y entornos de funcionamiento óptimos. Sin embargo se encuentran con el gran handicap de que no existen muchas testbed para su estudio físico, ni tampoco herramientas de software óptimas para su estudio mediante simulaciones. En este trabajo se presenta un modelo de simulación de OBS para el simulador de eventos discretos OMNeT++. Este modelo permite estudiar tanto los nodos frontera, como los nodos del core, así como enlazar la red OBS con otras redes de datos soportadas en OMNeT++, principalmente IP. Además, el diseño presenta una gran modularidad lo que permite modificar fácilmente el modelo OBS para incluir futuras propuestas que se hagan sobre esta tecnología.**

***Palabras Clave*—Optical Burst Switching, OBS, OMNeT++, simulación, arquitectura, diseño**

I. Introducción

Se ha propuesto Optical Burst Switching [1] (OBS) como un sistema de conmutación óptica con ventajas sobre las otras propuestas de conmutación óptica, como Optical Circuit Switching (OCS) y Optical Packet Switching (OPS). OBS tiene un gran interés entre los investigadores como la arquitectura óptica capaz de soportar una gran demanda de ancho de banda en backbones ópticos con Wavelength Division Multiplexing (WDM). OBS es una solución intermedia entre OCS y OPS. OBS tiene una gran utilización del ancho de banda, una baja latencia de establecimiento, sólo requiere de conmutación óptica de moderada velocidad, complejidad de procesamiento media y se adapta muy bien al tráfico intermitente.

En OBS la unidad de transporte básica es la ráfaga, que contiene un número de paquetes IP. Normalmente el agrupamiento de paquetes IP suele ser por destino, pero se puede hacer por cualquier criterio o combinación de criterios: dirección origen o destino, puertos de origen o destino, protocolos sobre IP, tipo de datos, etc. También se han propuesto diversos criterios para decidir cuándo enviar la ráfaga. Por ejemplo, se puede acumular todos los paquetes que lleguen durante un tiempo (con lo que su tamaño es variable) o se puede acumular exactamente una cantidad dada de bytes [2] (con lo que el tiempo para formar una ráfaga es variable). Las ráfagas se ensamblan y desensamblan en los nodos frontera o edge nodes, y cada formador de ráfagas se denomina burstifier. Una ráfaga se puede considerar como un paquete óptico que va desde el nodo entrada a la red OBS hasta el nodo salida de la red OBS sin sufrir ninguna conversión óptica-electrico-óptica (OEO). Para unir todas las parejas origen-destino y crear la red OBS se usan los nodos del core, que en el fondo son conmutadores ópticos o Optical Cross Connectors (OXC) gobernados por una Unidad de Control electrónica (UC).

En OBS se usa señalización fuera de banda con un tiempo de offset de retraso entre la señalización y la ráfaga. Así se obtiene separación espacial y temporal entre la ráfaga y el Burst Control Packet (BCP) [3], también denominado Control Packet Header (CPH). Esto es una característica específica de OBS que le da una gran capacidad de manejo y flexibilidad de red.

Generalmente OBS usa esquemas de señalización unidireccionales e iniciados por el origen, es decir, las ráfagas se envían a la red OBS sin esperar a la confirmación del éxito o del fracaso del intento de reserva del path completo hasta el destino. Así, las ráfagas pueden competir por los mismo recursos. Esta es la principal causa de la pérdida de ráfagas en OBS y sucede cuando el número de intentos simultaneos de reservas de ráfagas en una fibra de salida de un nodo del core es superior al número de longitudes de onda disponibles. También se puede dar que el BCP y su ráfaga asociada se acerquen tanto en tiempo, que el nodo del core no tenga tiempo a programar la conmutación y tenga que descartar la ráfaga. En cualquiera de los dos casos, si no se dispone de nodos del core con técnicas avanzadas como buffer ópticos, esto implica la pérdida de la ráfaga y que el rendimiento de OBS se vea afectado, bien en la utilización del ancho de banda o en la latencia por la retransmisión de la información perdida.

La Fig. 1 presenta un modelo de arquitectura de red OBS.

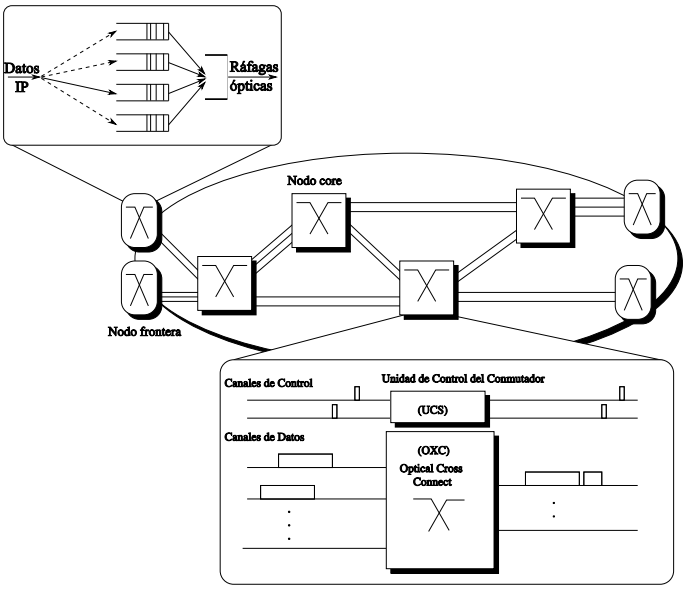


Figura 1. Esquema de arquitectura de red OBS

Este trabajo presenta un modelo de red OBS para el simulador OMNeT++[1] con el que estudiar esta nueva técnología. Actualmente existen propuestas teóricas de las diferentes partes de OBS (burstifiers por timer o tamaño, técnicas de señalización) y muy pocas implementaciones experimentales o testbed [4][5][6][7][8][9]. Por tanto se considera importante como primer paso poder estudiar todas estas propuestas usando herramientas de simulación.

Existen muchos simuladores de red [10]: el open source NS-2, el comercial OPNET, el específico para redes inhalambricas Qualnet, etc. Para este trabajo se ha escogido OMNeT++ [11]. A diferencia del resto su función principal no es la de simulador de redes, sino la de simulador genérico de eventos discretos con el que poder simular desde el funcionamiento de un disco duro hasta el comportamiento de una red Ethernet. Esto le da una gran versatilidad y capacidad de ser explotado en diversos ámbitos. OMNeT++ es de código abierto y tiene una *Academic Public License* que lo hace gratuito para usos no-comerciales. Está disponible para todas las plataformas comunes incluyendo Windows, Mac OS/X y Linux. Todo el código fuente está en C++ y se puede compilar con gcc o con el compilador Microsoft Visual C++.

Existen otros dos simuladores que también podrían ser los escogidos para este trabajo: OPNET[2] y NS-2[3].

OPNET tiene una licencia anual bastante cara, incluso para investigación científica, mientras que como se ha comentado OMNeT++ es gratuito para este fin. Además, pagando la licencia sólo se tiene acceso al código fuente de los modelos, pero no del kernel del simulador. Una diferencia significativa es que los modelos de OPNET son siempre de topología fija, mientras que en OMNeT++ es fácil tener topologías parametrizadas. En OPNET la manera habitual y preferida de definir la topología de red es mediante su editor gráfico, que guarda la red en un formato binario propietario que dificulta la posibilidad de generar estas topologías mediante programación (hay que usar una API específica para C de OPNET). En cambio en OMNeT++ las topologías se guardan en ficheros de textos planos que son fáciles de manipular. La principal ventaja de OPNET sobre OMNeT++ es su gran librería de modelos de protocolos (incluido uno para OBS), mientras que su naturaleza cerrada hace que la programación y la solución de problemas sea difícil.

NS-2 es el simulador de red más usado en el ámbito académico, pero no tiene la separación entre kernel de simulación y modelos que tiene OMNeT++: la distribución NS-2 contiene los modelos junto a la infraestructura del simulador como una única entidad inseparable. El objetivo de NS-2 es crear un simulador de red, mientras que el objetivo de OMNeT++ es ofrecer una plataforma de simulación. A NS-2 le faltan muchas herramientas y componentes de infraestructura que OMNeT++ tiene: soporte para modelos jerárquicos, interfaz gráfica de usuario (GUI) de entorno de simulación, separación entre modelos y experimentos, herramientas gráficas de análisis, algunas funcionalidades de simulación como multiples streams RNG, etc. Todo esto debido a que NS-2 se ha centrado en desarrollar los modelos de simulación, en vez de la infraestructura de simulación. Además, aunque NS-2 es open source y multiplataforma, en Windows pierde ciertas funcionalidades y es necesario compilar y usarlo mediante Cygwin[4], un entorno Linux para Windows.

Hasta ahora, aunque sí se ha propuesto algún modelo de simulación OMNeT++ para OBS [12], no se han hecho públicos. En cambio sí existen modelos OBS públicos para otros simuladores, como NS-2 [13]. Por esto se han realizado los módulos OMNeT++ necesarios (accesibles vía web[5]) para poder estudiar en profundidad las diferentes propuestas de la tecnología OBS. La arquitectura del modelo OBS de este trabajo es diferente al presentado en [12], basado en la propuesta teórica de [14], puesto que asume que existen dos tipos de nodos: los nodos frontera (edge node) y los nodos del core (core node). El primero, el nodo frontera, se encarga de introducir (y extraer) tráfico óptico a la red OBS usando los burstifiers para crear las ráfagas, pero no tiene tráfico óptico en tránsito pasando por él. El segundo, el nodo del core, sólo se encarga de tráfico óptico en tránsito, sin tener capacidad de introducir (o extraer) tráfico óptico de la red.

OMNeT++ provee de la maquinaria y las herramientas básicas para escribir esos componentes y esas simulaciones, en vez de proveer de componentes de simulación para redes de computadores, colas de redes y otros dominios. Es un framework, más que un programa de simulación. Para cada área específica de aplicación se han desarrollado modelos específicos, como el INET Framework para simulación de redes IP. El desarrollo de estos modelos es completamente independiente de OMNeT++, incluyendo sus ciclos de publicación, hasta tal punto que muchos son desarrollados por equipos externos a OMNeT++, aunque se publiquen en la misma página web de OMNeT++.

Un modelo OMNeT++ consiste en módulos que se comunican pasándose mensajes. Estos mensajes se pasan por las conexiones que se crean entre las **puertas**, o interfaces de entrada/salida, de los módulos. Los módulos básicos se denominan **módulos simples**. Los módulos simples se escriben en C++ usando las clases de la librería de simulación. Los módulos simples se pueden agrupar en **módulos compuestos**, Fig. 2, y estos en otros, hasta obtener la complejidad necesaria para describir el componente deseado. Los módulos compuestos se describen mediante el lenguaje NED usando ficheros de texto plano. Por ejemplo, cada interfaz de un router puede ser un módulo simple y el router sería un módulo compuesto. En OMNeT++ las redes son un tipo particular de módulos compuestos sobre las que se pueden realizar simulaciones.

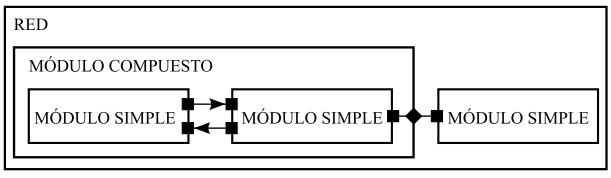


Figura 2. Estructura de módulos en OMNeT++

La información que se pasa entre los módulos y los eventos programados para el simulador se describe mediante **mensa-**

[1] http://www.omnetpp.org
[2] http://www.opnet.com/
[3] http://nsnam.isi.edu/nsnam/
[4] http://www.cygwin.com/
[5] **https://www.tlm.unavarra.es/investigacion/proyectos/strrong/soft/**

jes. Un mensaje es una estructura C++ específica de OMNeT++ que contiene variables para describir la información deseada. Por ejemplo, mediante un mensaje se puede describir un paquete IP, en el que cada variable de la estructura es un campo de la cabecera IP. Los mensajes se describen mediante un fichero de texto y se pueden ampliar para describir cualquier dato. OMNeT++ preprocesa la estructura de definición del mensaje generando automáticamente el código C++ con el que poder manejar el mensaje mediante una función *setter* y una *getter* para cada variable de la estructura.

A la hora de usar un módulo, el usuario no sabe si es un módulo simple o compuesto, sólo necesita conocer cómo se tienen que conectar sus puertas y qué parámetros necesita. Así, se puede ampliar el módulo subdiviendolo internamente en módulos simples o simplificarlo pasandolo a un único módulo simple, sin que el usuario final y las simulaciones ya preparadas para usar esos módulos se den cuenta del cambio.

El resto del paper se organiza de la siguiente manera. La sección II presenta el modelo OBS que se va a implementar en OMNeT++. En las secciones III, IV y V se explica pormenorizadamente la implementación realizada del modelo OBS, remarcando su versatilidad y modularidad. En la sección VI se presenta un ejemplo de uso de los módulos OBS desarrollados. Por último, la sección VII presenta las conclusiones.

II. El modelo de red OBS

Tanto el nodo frontera como el nodo del core se implementan en módulos compuestos de OMNeT++: ***OBS_edgeNode*** y ***OBS_CoreNode***. En las siguientes secciones se presentarán desde el punto de vista funcional, remarcando su capacidad de configuración y su modularidad.

III. Nodo Frontera - módulo OBS_EdgeNode

A la hora de implementar un nodo frontera, este se puede ver como un router al que por lo menos se le ha añadido una interfaz OBS, o como un router que a su salida se le ha conectado un equipo OBS. Cualquiera de los dos casos son factibles y fáciles de hacer en OMNeT++. En este modelo de simulación se ha implementado el primer caso, el router con interfaz OBS, basándose en los módulos del router básico de OMNeT++.

Cuando actúa como nodo de ingreso, módulo ***assembler***, el nodo frontera es responsable de tomar la decisión de enrutamiento, agregar o ensamblar el tráfico entrante en ráfagas y planificar la transmisión de la ráfaga en el canal de salida.

En OBS el tráfico IP de entrada se agrega a ráfagas en el nodo frontera según el destino óptico. Esta agregación se realiza en los burstifiers y por lo menos tiene que haber un burstifier por destino (o nodo frontera de salida). Se puede tener más de un burstifier por destino para poder diferenciar el tráfico (y las ráfagas que se generan) dependiendo de las direcciones IP de origen o destino, del puerto de origen o destino y del protocolo IP usado. En el modelo propuesto en [12] no se le da importancia a la manera de formar las ráfagas, puesto que sólo se quiere estudiar el comportamiento del backbone OBS. Pero este planteamiento no permite usar el modelo OBS en simulaciones que conecten la red OBS con otras redes de datos. El modelo propuesto en este trabajo sí permite hacer esto. Además, así es posible estudiar el tráfico que componen las ráfagas y cómo cambia esta generación dependiendo del tráfico de entrada y los esquemas de agregación usados en los burstifiers. La implementación actual del nodo frontera soporta los esquemas más extendidos: timer, tamaño [2] y número de paquetes [15]. Y, por supuesto, la mezcla de estos esquemas. Pero añadir un nuevo esquema sólo implica la modificación de un único módulo simple de OMNeT++, el ***packetBurstifier***.

El forwarding óptico se hace mediante un esquema tipo *conmutación óptica de etiquetas*, similar a LOBS [16]. Cada ráfaga lleva una etiqueta. En los nodos del core se usa esta etiqueta, junto con el puerto y la longitud de onda por la que ha llegado, como parámetros de forwarding. Las etiquetas pueden cambiar en cada salto o nodo del core. La etiqueta inicial la pone el burstifier del nodo frontera que genera la ráfaga. La elección de qué burstifier genera la ráfaga se hace en dos pasos. Primero se decide la interfaz óptica por la que el paquete IP puede llegar a su destino. Para eso se usa la tabla de rutas del nodo frontera. Una vez decidido el interfaz óptico, el ***dispatcher*** de esa interfaz, mediante sus criterios de clasificación, decide en qué burstifier almacenar el paquete IP. La etiqueta que se le da a la ráfaga generada es la etiqueta que se configura en la simulación a ese burstifier.

Una vez generada la ráfaga en el burstifier, esta se pasa al nivel de enlace OBS. Este nivel de enlace se ha implementado como una cola en la que almacenar las ráfagas hasta su transmisión. El tamaño de esta cola, tanto en bits como en número de ráfagas, es configurable para cada simulación y nodo frontera. Cuando una ráfaga no cabe en la cola, simplemente se descarta.

Para planificación actualmente se usa el esquema más básico propuesto, denominado **Horizon o LAUC** [17]. En este esquema, cuando la ráfaga llega a la cola se calcula cuál es el horizonte más cercano en que cualquiera de las longitudes de onda queda libre. Se planifica para ese instante y esa longitud de onda, y se modifica el horizonte de la longitud de onda. Para que los nodos del core tengan tiempo de procesar la información de control antes de que llegue la ráfaga, el BCP asociado a la ráfaga se envía antes. El offset temporal entre el BCP y la ráfaga tiene un máximo y un mínimo. Inicialmente el envío del BCP se planifica para ese offset máximo. Si por congestión en el canal de control, este no puede salir del nodo frontera con el offset mínimo de separación, el BCP y la ráfaga se replanifican.

Cuando actúa como un nodo de salida, módulo ***disassembler***, el nodo frontera realiza la operación inversa, es decir, desagrega las ráfagas en paquetes y los procesa como cualquier otro tráfico.

En la Fig. 3 se muestra la estructura interna del nodo frontera. Como ya se ha comentado anteriormente el nodo frontera se basa en el router simple de OMNeT++, añadiéndole una interfaz OBS. Así, igual que en el router simple, tenemos un ***interfaceTable*** con la descripción y parámetros de red de las interfaces de red, y un ***routingTable*** con la tabla de rutas del router. En el ejemplo usado para la figura 3, el nodo frontera tiene una interface Ethernet para que una red de conmutación de paquetes acceda al backbone OBS, y una única interfaz OBS con la que conectarse con un nodo del core. Para cada simulación y nodo frontera, tanto la tabla de rutas como el

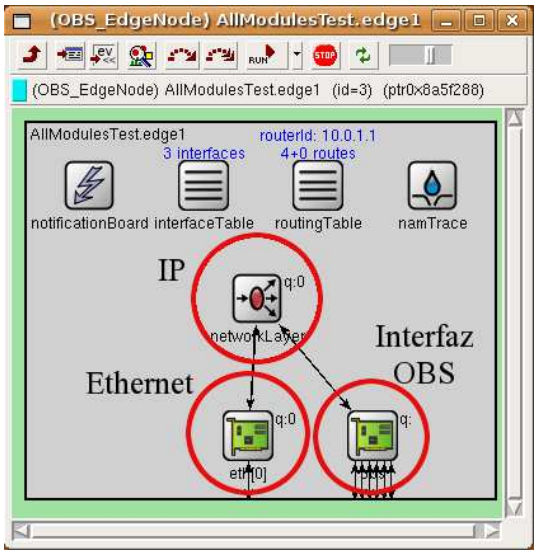


Figura 3. Nodo Frontera

número de interfaces OBS son faćilmente configurables.

La interfaz OBS es la responsable de hacer todo el trabajo relacionado con la parte OBS. Los parámetros interesantes son configurables independientemente para cada interfaz OBS. La red tiene que mantener cierta coherencia, p.e. no se puede conectar un puerto de salida del nodo frontera de 5 longitudes de onda con un puerto de entrada del nodo del core de sólamente 3 longitudes de onda. OMNeT++ considera cada longitud de onda de una fibra óptica como un enlace independiente, por lo que las dibuja por separado. Esto se ve en las figuras 3 y 4, donde sólo hay una fibra óptica de 5 longitudes de onda (4 de datos y una de señalización).

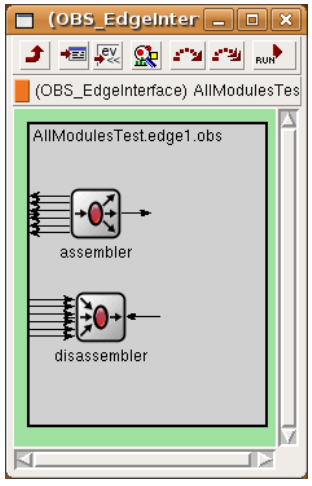


Figura 4. Interfaz OBS del Nodo Frontera

La interfaz OBS se implementa mediante los siguientes 2 submodulos:

III-A. assembler

Es un módulo complejo que realiza el trabajo relacionado con el funcionamiento del nodo frontera como nodo de ingreso. Se encarga de tomar la decisión de enrutamiento, ensamblar el tráfico entrante en ráfagas y planificar la ráfaga. El ***assembler***, como se puede ver en la figura 5, está compuesto por tres módulos simples.

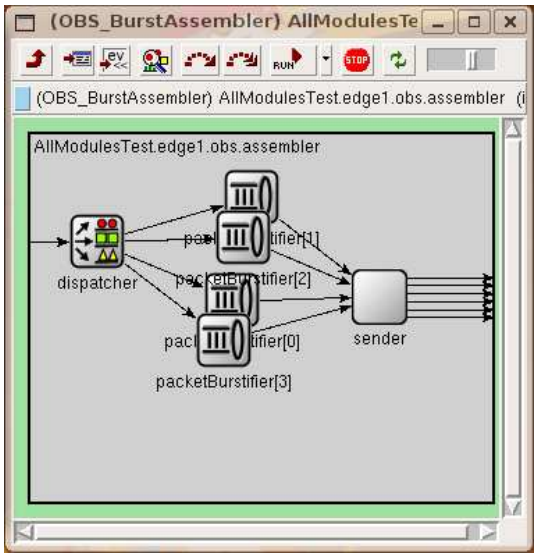


Figura 5. Assembler del Nodo Frontera

1. ***dispatcher - clasificador***
   En base a las reglas de clasificación, se encarga de enviar a cada burstifier el tráfico entrante que le corresponde. Las reglas de clasificación de cada ***dispatcher*** se configuran en un fichero que se le pasa en la inicialización. En el fichero de reglas de cada nodo frontera tiene que haber una única regla para cada uno de sus burstifier. Las reglas pueden tener tantas parejas *parámetro-valor* como tenga que cumplir el tráfico que va dirigido a ese burstifier. El tráfico entrante que no cumpla ninguna regla se descarta. Por ejemplo, se puede hacer una regla para agrupar en el mismo burstifier todo el tráfico (TCP o UDP) que vaya a la dirección 10.1.1.1 y puerto 80.
   Este módulo guarda estadísticas de los tamaños de los paquetes y el número de paquetes descartados.

2. ***packetBurstifier***
   Se encarga de ensamblar el tráfico entrante en una ráfaga usando el esquema de ensamblado y los parámetros que se le han configurado. Tiene parámetros para poder manejar los diferentes esquemas de ensamblado. Así, se puede configurar el tiempo máximo de ensamblado de una ráfaga, *timeout*, o el tamaño máximo de la ráfaga, *maxSize*. Pero también el tamaño de la cabecera, el tamaño mínimo de la ráfaga para un burstifier por *timeout* o si en un burstifier por *maxSize* el paquete que hace superar este límite se incluye en esa ráfaga o en la siguiente. Además, para dar mayor libertad, también se puede configurar el tiempo de offset máximo y mínimo para que sea diferente en cada burstifier.
   Este módulo guarda estadísticas del tiempo medio que pasan los paquetes en el burstifier y el número de paquetes por ráfaga.
   El planteamiento modular que se ha usado permite que en caso de querer incluir un nuevo esquema de

ensamblado, el único módulo a modificar sea este.

3. ***sender - nivel de enlace***
   Es el módulo encargado de planificar y transmitir la ráfaga y su BCP asociado. En este módulo se puede configurar tanto el tamaño de los BCPs como el tamaño de la cola del sistema donde almacenar las ráfagas mientras esperan su instante de transmisión. Es el único módulo del ***assembler*** que necesita saber el número de longitudes de onda de la fibra y su velocidad de transmisión.

III-B. disassembler

Es un módulo simple, que realiza el trabajo relacionado con el funcionamiento del nodo frontera como nodo de salida. Recibe las ráfagas, las desagrega y pasa los paquetes al nivel superior para que se procesen como cualquier otro tráfico.

IV. Nodo del Core - módulo OBS_CoreNode

Los nodos del core son responsables del procesamiento de los BCPs, de la conmutación de las ráfagas de una fibra de entrada a otra de salida sin conversión electro-óptica, y del mecanismo de contienda entre ráfagas. En OBS la señalización típicamente se hace fuera de banda. Normalmente transmitiendo el BCP asociado a una ráfaga en una longitud de onda exclusiva y diferente de las longitudes de onda para las ráfagas. En esta implementación se ha escogido usar para ello la primera longitud de onda de todas las fibras. En caso de querer modificar esto, sólo sería necesario modificar los módulos simples de entrada/salida: el ***Input*** y ***Output*** en el nodo del core y el ***sender*** en el nodo frontera.

Se han propuesto diferentes esquemas de señalización [18] [19], pero los protocolos más populares de señalización distribuida en OBS son Just-in-Time (JIT) [20] y Just-Enough-Time (JET) [21]. Los dos son esquemas de señalización unidireccionales e iniciados por el origen, es decir, las ráfagas se envían a la red OBS sin esperar a la confirmación del éxito o del fracaso del intento de reserva de un path hacia el destino. Se basan en el mismo principio, pero se diferencian en la duración de las reservas. JIT usa reserva inmediata desde que el BCP llega al nodo del core, mientras que JET retrasa la reserva del canal hasta la llegada estimada de la ráfaga. Como la información de señalización necesaria es diferente, en JET el BCP tiene que indicar cuándo se espera que llegue la ráfaga, el tipo de BCP usado en JET y JIT es diferente.

El retraso de la reserva hace que JET sea más eficiente que JIT, obteniendo mejores ratios de bloqueo y retrasos extremo-a-extremo más pequeños [14]. Por esta razón la implementación actual de este modelo de simulación OBS usa un esquema tipo JET, aunque se podría cambiar fácilmente a un esquema tipo JIT. Sólo sería necesario cambiar una pequeña parte del nodo del core, el módulo simple ***ControlUnit***. No sería necesario modificar el mensaje BCP porque el actual ya tiene todos los campos necesarios para señalizacion JIT.

Según el puerto, longitud de onda y etiqueta de entrada de la ráfaga, esta tiene asociadas en el ***ControlUnit*** unas longitudes de ondas que puede usar o *válidas* para cada puerto de salida. A la llegada del BCP, se escoge entre las longitudes de onda de salida *válidas* para su ráfaga asociada la que tenga el horizonte más cercano y menor del instante previsto de llegada de la ráfaga. Se planifica el ***Optical Cross-Connect (OXC)*** para que en ese instante de llegada conmute la longitud de onda de entrada con la longitud de onda de salida escogida y deshaga la conmutación una vez pasada la ráfaga. Actualmente el esquema asume que siempre hay un conversor de longitud de onda disponible para hacer esta conmutación entre longitudes de onda. Si no hay ninguna longitud de onda libre para ese instante de llegada, entonces simplemente el BCP se descarta y cuando la ráfaga llega, como su entrada no está conectada a ninguna salida, la ráfaga se pierde.

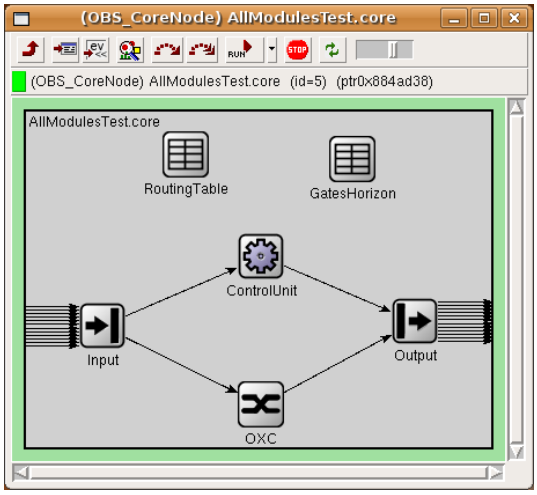


Figura 6. Nodo del Core

En la Fig. 6 se muestra la estructura interna del nodo del core. Se basa en realizar forwarding de las ráfagas de una fibra a otra mediante una matriz de conmutación dinámica configurable. Esta matriz de conmutación sigue las órdenes de la unidad de control. La unidad de control toma sus decisiones usando la *tabla de forwarding*, que indica el puerto y longitud de onda de salida según la etiqueta de la ráfaga. Actualmente esta tabla de forwarding se genera en la inicialización a partir de un fichero, y se mantiene inalterable durante la simulación. Se podría modificar la implementación del nodo del core para incluir una tabla de forwarding dinámica que cambiase su estado p.e. mediante un esquema de enrutamiento centralizado.

El nodo del core se compone de 4 submódulos: 3 módulos simples y uno compuesto.

IV-A. Input

Al haber escogido que la señalización (los BCPs) vayan en una longitud de onda exclusiva, es necesario separar de todas las fibras de entrada las longitudes de onda de señalización de las de transmisión. De esto se encarga el módulo ***Input***. Enlaza las longitudes de onda de señalización con la unidad de control (***ControlUnit***) y las de tráfico con la matriz de conmutación o ***Optical Cross-Connect (OXC)***.

IV-B. Output

Este módulo hace el trabajo inverso del módulo ***Input***: inyecta cada longitud de onda de señalización en su fibra correspondiente junto con el resto de longitudes de onda de tráfico.

Si en algún momento se desea modificar la longitud de onda que se usa para senalización OBS o usar señalización por medios alternativos como ADSL o enlaces de microondas, los módulos a modificar en el nodo del core serían estos dos. Tanto la matriz de conmutación como la unidad de control no se verían afectados y no sabrían que la señalización ha cambiado.

IV-C. OXC - Optical Cross-Connect

Es la matriz de conmutación óptica. Según le va indicando la unidad de control, conecta y desconecta las longitudes de onda de las fibras de entrada con las longitudes de onda de las fibras de la salida. El ***OXC*** no necesita saber qué tráfico pasa, cómo es la señalización OBS y ni siquiera si tiene completa capacidad de conversión de longitud de onda. Todo eso es responsabilidad y tarea de la unidad de control, por lo que este módulo es muy simple pero muy fácil de mantener ante cualquier cambio del nodo del core.

Se puede configurar el tiempo que necesita para hacer una conmutación. La unidad de control tendrá en cuenta este tiempo a la hora de planificar las conmutaciones, ya que para cuando llegue la ráfaga la conmutación tiene que estar completamente acabada o la ráfaga se corromperá.

IV-D. ControlUnit

La unidad de control es la encargada de realizar todo el procesamiento de la señalización y el forwarding y planificación de las ráfagas. Para eso se compone de tres módulos simples:

1. ***OE***
   El procesamiento de la señalización se realiza en el dominio electrónico, por lo que primeramente, como se ve en la Fig. 7 hace una conversión opto-eléctrica de los BCPs ópticos. En simulación esto sirve para emular el tiempo físico de procesamiento opto-electrónico, y poder añadir al mensaje BCP información adicional (puerto de entrada) que necesita la unidad de control.

2. ***ControlUnit***
   Con la información de los BCPs electrónicos y la tabla de forwarding busca entre las longitudes de onda *válidas* la que tenga el horizonte más cercano y menor al tiempo estimado de llegada de la ráfaga. Para el cálculo de este tiempo de llegada estimado, la unidad de control tiene en cuenta tanto el tiempo de conmutación mencionado anteriormente, como un tiempo de guarda configurable para cada nodo del core. Con este tiempo de guarda se puede configurar la separación mínima que tiene que haber entre dos ráfagas a conmutar hacia la misma longitud de onda de salida. El tiempo de procesamiento de la información de control es un parámetro configurable y constante por BCP.
   Las reglas de forwarding de cada nodo del core se cargan en la inicialización de un fichero. En este fichero se indica cual es la combinación de puerto, longitud de onda y etiqueta de salida para cada combinación de puerto, longitud de onda y etiqueta de entrada. Tanto para los puertos como para las longitudes de onda se puede usar el símbolo "*"para indicar cualquier fibra y cualquier longitud de onda. Con este sencillo fichero de configuración se puede implementar un forwarding tipo *conmutación óptica de etiquetas*.

3. ***EO***
   Realiza el proceso inverso del módulo ***OE***, la conversión electro-óptica de los BCPs electrónicos. Para ello, igual que el ***sender*** del nodo frontera, necesita conocer la velocidad de transmisión de las longitudes de onda.

Por cada BCP óptico la unidad de control tiene que hacer una conversión opto-eléctrica, procesar la información del BCP y una conversión electro-óptica. El BCP óptico tarda más tiempo en atravesar el nodo del core que la ráfaga óptica, por lo que el offset temporal entre ellos se acorta. En redes OBS con muchos saltos, el offset se puede hacer tan pequeño que sea imposible procesar el BCP antes de la llegada de la ráfaga y esta se pierde. Para evitar esta situación se pueden emplear esquemas como ODD [22], que usan líneas de retardo (FDL) en los nodos del core para mantener el offset constante. El simulador actualmente utiliza un modelo clásico sin FDLs, pero para soportar esto simplemente se deberían de incluir las FDLs en el ***OXC*** y que la unidad de control, al conmutar el ***OXC***, establezca también el tiempo a retrasar la ráfaga.

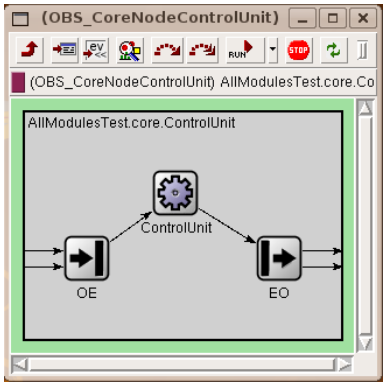


Figura 7. Unidad de control del Nodo del Core

V. Otros módulos

Por último es necesario mencionar que aunque con los módulos ***OBS_edgeNode*** y ***OBS_coreNode*** se puede realizar cualquier topología OBS, se han implementado otros dos componentes OBS para su uso en simulaciones: un sniffer óptico y una módulo de pérdidas.

1. ***OpticalMonitor***
   Es la implementación en OMNeT++ de un sniffer óptico. Su función es recoger estadísticas detalladas de las ráfagas que pasan por él, sin alterar de ninguna manera las ráfagas ni el funcionamiento de la red OBS.
2. ***DropBurst***
   Se usa para simular el efecto de que un segmento de red OBS pierde parte de las ráfagas que pasan por él. Actualmente sólo simula pérdidas independientes identicamente distribuidas, pero se puede modificar para que por ejemplo simule pérdidas que dependan del tamaño de las ráfagas.

VI. Simulaciones

Para demostrar el funcionamiento del simulador se ha realizado una red con la topología que se ve en la Fig. 8. Une cuatro nodos frontera usando dos nodos del core y cinco enlaces bidireccionales. Los enlaces son fibras ópticas con ocho longitudes de onda de datos por enlace y con 1Gbps de capacidad de transmisión por longitud de onda. Los nodos frontera están configurados para acumular tráfico durante 5ms (burstifier por timer) y tienen un buffer electrónico infinito.

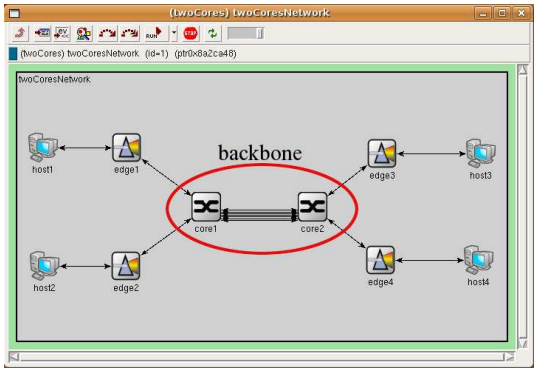


Figura 8. Red usada en las simulaciones

Hay dos flujos de tráfico UDP: tráfico de interés del *host1* al *host3* y tráfico interferente del *host2* al *host4*. En los dos casos el tiempo entre llegadas sigue una distribución exponencial y los tamaños de los paquetes son constantes. El tráfico de interés tiene un bitrate medio de 1.6Gbps, o lo que es lo mismo, el 20 % de la capacidad del enlace entre los dos nodos del core. Se ha hecho un barrido en el bitrate medio del tráfico interferente para que ocupe desde 2.5 % al 90 % de la capacidad del enlace entre los dos nodos del core. Como los nodos frontera tienen buffer electrónico infinito, las ráfagas sólo se pueden perder por contienda en el nodo del core *core1*.

En la figura 9 se puede ver la probabilidad de pérdida de ráfagas en la red OBS. Como era de esperar, se ve que no hay pérdidas por contienda en el nodo del core *core1* hasta valores medios de ρ del tráfico interferente. A partir de ese punto las pérdidas siguen aumentando.

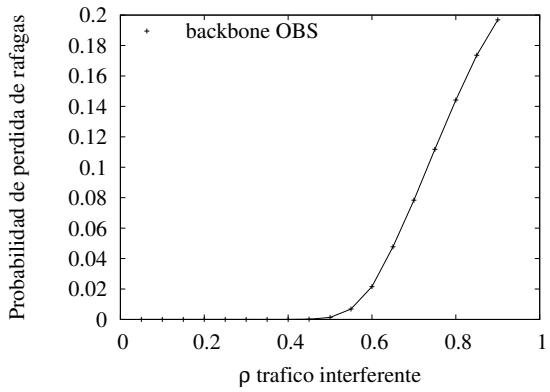


Figura 9. Probabilidad de pérdida de ráfagas

VII. Conclusiones

En este trabajo se ha presentado un modelo de simulación de OBS para el simulador de eventos discretos OMNeT++. Este modelo incluye tanto la implementación del nodo frontera como la del nodo del core teniendo siempre en mente la modularidad que permita incluir futuras propuestas que se hagan sobre esta tecnología.

Se ha visto que es fáctible la realización de un modelo de simulación de OBS para su uso en investigación. Aunque sea el primer desarrollo de este modelo, simula correctamente el funcionamiento básico de OBS.

En un futuro se prevee incluir mejoras. En los nodos frontera se pueden incluir nuevos esquemas de agregración y esquemas de planificación más avanzados que el **Horizon o LAUC**. En el nodo del core se puede plantear usar una *tabla de forwarding* dinámica o la inclusión de líneas de retardo (o FDLs) para darle cierta capacidad de buffering.

Agradecimientos

Este trabajo ha sido financiado por el MEC (proyecto STRRONG TEC2007-62192/TCM). Los autores quieren agradecer a la red temática IPoTN (TEC2008-02552-E/TEC).

Referencias

[1] C. Qiao and M. Yoo, "Optical Burst Switching (OBS) - A New Paradigm for an Optical Internet," *Journal of High-Speed Networks*, vol. 8, no. 1, 1999.

[2] J. Choi, J. Choi, and M. Kang, "Dimensioning Burst Assembly Process in Optical Burst Switching Networks," *IEICE Transactions on Communications*, vol. E88-B, pp. 3855–3863, October 2005.

[3] T. Battestilli and H. Perros, "An introduction to optical burst switching," *Communications Magazine, IEEE*, vol. 41, pp. S10–S15, Aug. 2003.

[4] A. Campi, W. Cerroni, F. Callegati, G. Zervas, R.Ñejabati, and D. Simeonidou, "SIP Based OBS networks for Grid Computing," in *LNCS 4534, Proceedings of ONDM*, (Athens, Greece), May 2007.

[5] W. Zhang, J. Wu, J. Li, W. Minxue, and S. Jindan, "TCP Performance Experiment on LOBS Network Testbed," in *LNCS 4534, Proceedings of the 11th International IFIP TC6 Conference, ONDM* (I. Tomkos, F.Ñeri, J. Sole, X. Masip, and S. Sanchez, eds.), (Athens, Greece), pp. 186–193, May 2007.

[6] J. Kim, J. Cho, M. Jain, D. Gutierrez, L. Kazocsky, C. Su, R. Rabbat, and T. Hamada, "Demonstration of a 2.5 Gbps Optical Burst Switched WDM Ring Network," in *Proceedings of OFC/NFOEC*, March 2006.

[7] Y. Sun, T. Hashiguchi, V. Minh, X. Wang, H. Morikawa, and T. Aoyama, "A Burst-Switched Photonic Network Testbed: Its Architecture, Protocols and Experiments," *IEICE Transactions on Communicactions*, vol. E88-B, pp. 3864–3873, October 2005.

[8] Z. Gao, Y. Qiao, Y. Ji, and T. Saito, "The Optical Burst Switching Ring Network Using AOTF to Drop Data Burst," *Journal of Optical Communications*, vol. 26, no. 6, pp. 255–259, 2005.

[9] Y. Sun, T. Hashiguchi, V. Minh, X. Wang, H. Morikawa, and T. Aoyama, "Design and Implementation of an Optical Burst-Switched Network Testbed," *IEEE Communications Magazine*, vol. 43, pp. S48–S55, November 2005.

[10] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Simutools '08: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, (ICST, Brussels, Belgium, Belgium), pp. 1–10, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[11] A. Varga, "The OMNET++ discrete event simulation system," in *Proceedings of the European Simulation Multiconference*, (Prague, Czech Republic), pp. 319–324, SCS – European Publishing House, June 2001.

[12] A. L. Barradas and M. C. R. Medeiros, "An OMNeT++ model for the evaluation of OBS routing strategies," in *Simutools '08: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, (ICST, Brussels, Belgium, Belgium), pp. 1–7, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[13] Optical Internet Research Center, "OIRC OBS-ns simulator," April 2008. http://wine.icu.ac.kr/~obsns/.

[14] J. P. Jue and V. M. Vokkarane, *Optical Burst Switched Networks*. Springer, 2005.

[15] X. Yu, Y. Chen, and C. Qiao, "A Study of Traffic Statistics of Assembled Burst Traffic in Optical Burst Switched Networks," in *Proceedings of SPIE Opticomm*, (Boston), pp. 149–159, SPIE, July 2002.

[16] C. Qiao, "Labeled optical burst switching for IP-over-WDM integration," *Communications Magazine, IEEE*, vol. 38, pp. 104–114, Sep 2000.
[17] J. S. Turner, "Terabit burst switching," *J. High Speed Netw.*, vol. 8, no. 1, pp. 3–16, 1999.
[18] J. Teng and G.Ñ. Rouskas, "A comparison of the JIT, JET, and Horizon wavelength reservation schemes on a single OBS node," in *In Proc. of the First International Workshop on Optical Burst Switching*, p. 2003, 2003.
[19] A. Zalesky, E. Wong, M. Zukerman, H. L. Vu, and R. Tucker, "Performance analysis of an OBS edge router," *Photonics Technology Letters, IEEE*, vol. 16, pp. 695–697, Feb. 2004.
[20] J. Wei and J. McFarland, R.I., "Just-in-time signaling for WDM optical burst switching networks," *Lightwave Technology, Journal of*, vol. 18, pp. 2019–2037, Dec 2000.
[21] Y. Chen, C. Qiao, and X. Yu, "Optical burst switching: a new area in optical networking research," *Network, IEEE*, vol. 18, pp. 16–23, May-June 2004.
[22] L. Xu, H. G. Perros, and G.Ñ. Rouskas, "A simulation study of optical burst switching and access protocols for wdm ring networks," *Comput. Netw.*, vol. 41, no. 2, pp. 143–160, 2003.

Contribuciones al análisis multiresolución de matrices de tráfico

David Rincón Rivera, Javier Torres Haba
Departamento de Ingenieria Telemática (ENTEL) – Escola Politècnica Superior de Castelldefels (EPSC)
Universitat Politècnica de Catalunya (UPC)
Edifici C4, C/ Esteve Terrades,7, Castelldefels 08860 (Barcelona)
drincon@entel.upc.edu, jose.javier.torres-haba@estudiant.upc.edu

***Resumen*- Las matrices de tráfico (TM) constituyen una información muy relevante para los operadores de red. Gran parte de las investigaciones sobre TMs se ha centrado en su inferencia a partir de medidas indirectas, pero no en otros aspectos importantes como su síntesis, detección de anomalías o predicciones fiables de tráfico. Estas aplicaciones requieren un buen modelo de la TM. En el presente artículo se aborda la búsqueda de un modelo general, con la convicción de que deberá ser disperso (*sparse*); es decir, que tenga pocos parámetros en comparación al tamaño de la matriz. El Análisis Multi-Resolución (MRA) es una técnica que nos puede ayudar a encontrar dicha representación dispersa. En este trabajo se propone el uso de la transformada wavelet de difusión (DW), ya que se adapta inherentemente a la topología del grafo subyacente. El documento describe la construcción de una versión en 2D de la transformada DW y muestra su aplicación en el análisis de matrices de tráfico reales obtenidas en redes operacionales. Los resultados preliminares confirman la compresibilidad de las TMs en pocos parámetros y su posible aplicabilidad a otras tareas relacionadas con estas TMs.**

***Palabras Clave*- Caracterización de Tráfico, Matrices de Tráfico, Wavelets de Difusión, Análisis Multi-Resolución.**

I. INTRODUCCIÓN

Dada una red de conmutación de paquetes de *N* nodos (típicamente routers), definimos su matriz de tráfico (*traffic matrix*, TM) como el volumen de tráfico enviado desde uno de los nodos a cualquiera de los otros. En principio las diagonales de la matriz estarían vacías, pero cuando los nodos son PoPs (*Points of Presence*) en vez de routers, y dado que un PoP no sólo conmuta tráfico proveniente de los PoP vecinos sino que también da servicio a redes locales que pueden intercambiar tráfico entre ellas, la TM tiene tráfico en las diagonales (es decir, del nodo *i* hacia el nodo *i*). El volumen de tráfico se mide durante un cierto intervalo temporal (típicamente del orden de algunos minutos u horas). La Fig. 1 muestra la topología de la red Abilene y un ejemplo de matriz de tráfico de la misma red.

Dada su importancia en las tareas de ingeniería de red (monitorización de la carga, detección de anomalías, predicción de saturación de los enlaces y provisión de red, etc.), el estudio de las TM es de gran interés [1]. Sin embargo, son difíciles de medir directamente [2], por lo que muchos investigadores han abordado su deducción indirecta (inferencia) a partir de otras medidas como por ejemplo la carga de los enlaces, que es fácilmente obtenible a partir de las MIB SNMP presentes en las interfaces de red. Se han conseguido algunos resultados prácticos, como por ejemplo en ingeniería de tráfico [3]. Sin embargo, estas medidas no

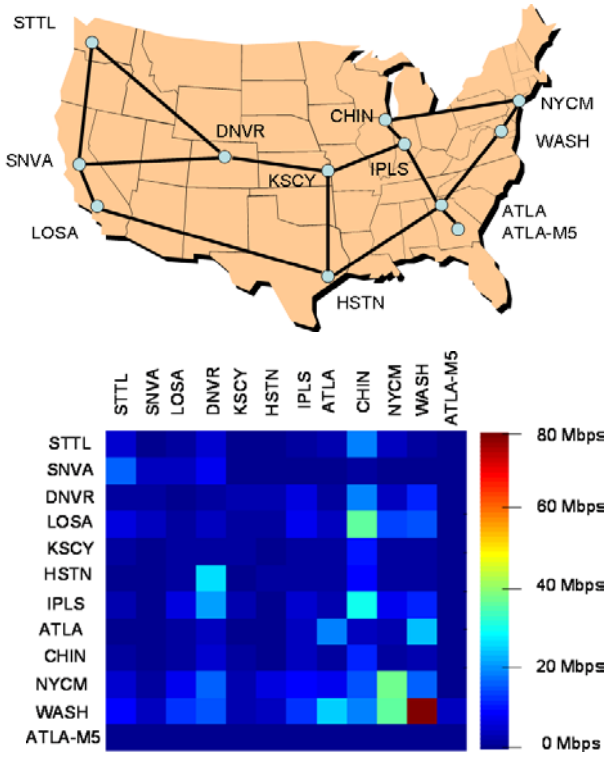


Fig. 1: Arriba: topología de los PoPs de Abilene en 2004. Abajo: matriz de tráfico de Abilene del día 3 de marzo de 2004 de 12:00 a 12:05.

nos dan suficiente información para recrear la TM original, por lo que se necesitan datos adicionales. Uno de esos datos puede ser un modelo *a priori* de la matriz de tráfico [4,5,6,7,8].

El interés de los operadores de redes IP en conseguir una buena estimación de la TM es debido a la aplicación práctica en sus operaciones diarias. En este sentido un aspecto importante es la síntesis de TMs [9, 10], que nos permite crear matrices de demanda que son el punto de partida para el diseño de la topología de nuevas redes (localización de los PoPs/routers, enlaces entre nodos y su velocidad). Es muy importante que las particularidades de las matrices queden reflejadas en la síntesis con la máxima fidelidad posible para minimizar el (posible) error del diseño. No obstante, muy pocos estudios han abordado las propiedades de las TMs. Además, un aspecto que no facilita en absoluto esta tarea de investigación es que prácticamente ningún ISP comercial hace públicos los datos reales correspondientes a sus

medidas, lo que dificulta tanto el desarrollo de modelos de TM como su validación. Las redes financiadas con fondos públicos (GÉANT, Abilene) ofrecen algo más de colaboración (véase [11]).

Otra aplicación importante de las TMs en la ingeniería de redes es su uso en técnicas de ingeniería de tráfico; por ejemplo, para balancear las cargas sobre los enlaces de una red. Por eso sería ideal encontrar un modelo que nos sirviese para hacer predicciones de la demanda de tráfico a corto y medio plazo. Finalmente, y como subproducto del modelo predictivo, también se pueden detectar anomalías en la red (comportamientos alejado de la predicción propuesta) que pueden corresponder a cambios en los patrones de uso de la red, o bien a un ataque malicioso de denegación de servicio, por poner dos ejemplos concretos.

En todos los casos anteriores necesitamos un buen modelo de matriz de tráfico; es decir, expresar sintéticamente la TM en función de unos (pocos) parámetros. En nuestra opinión, una de las cualidades que debe cumplir un buen modelo es que sea disperso (*sparse*), en el sentido de tener un número de parámetros mucho menor que el número de elementos de la matriz. Una matriz de tráfico para una red con N nodos tiene N^2 elementos, y, teniendo en cuenta que N puede ser del orden de un millar de nodos, obtendríamos matrices muy grandes. Un modelo disperso debería tener un número de parámetros $M<<N^2$. Hay varias razones para escoger un modelo de este tipo [23]:

- En general, hay un compromiso entre la fidelidad del modelo y su capacidad predictiva: si tiene muchos parámetros obtendremos un modelo muy fiable para ese conjunto de datos, pero que no nos servirá como modelo predictivo porque será demasiado específico y será difícilmente aplicable a otras situaciones.
- Si el modelo dispone de pocos parámetros será mucho más fácil asignarles sentido físico, de manera que se incrementará el control que tenemos sobre el modelo.
- El problema de la inferencia de las TMs a partir de las medidas SNMP radica en que se tienen K medidas (del orden de N) y N^2 parámetros que deducir; esto es, más incógnitas que ecuaciones. Si el modelo disperso tiene $M \leq K$ parámetros, quizá sea posible obtener una solución.

Existen algunos modelos dispersos, como el modelo de gravedad [7], que tiene tan sólo $2N$ parámetros ($2N << N^2$). Sin embargo, este modelo no es necesariamente una buena representación de la TM, sino que más bien es un modelo *a priori* que luego es corregido mediante la proyección del modelo de gravedad hacia la solución más cercana de las pertenecientes al espacio de soluciones posibles de la TM. Hasta ahora la búsqueda de modelos fiables no ha partido de este concepto de dispersión; la mayoría se ha justificado a través de la prueba empírica de comparar los resultados con los datos obtenidos de la medidas.

Nuestra aportación a la búsqueda de ese modelo disperso es el uso de la técnica denominada Análisis Multi-Resolución (*Multi-resolution Analysis*, MRA). Intuitivamente, podemos considerar el MRA como una manera de observar los datos a diferentes resoluciones o escalas, como si fuera un *zoom*: si miramos de cerca, vemos muchos detalles (alta frecuencia), pero si nos alejamos, obtenemos una visión global con pocos detalles (bajas frecuencias) [14]. Normalmente el MRA se realiza mediante la transformada wavelet, y se aplica a la compresión mediante la eliminación de las altas frecuencias, como es el caso del estándar de compresión de imagen JPEG 2000 [12]. En general MRA se puede entender como un decorrelador, en el sentido de reducir la representación de la señal a unos pocos parámetros en el dominio transformado, obteniendo así una señal transformada dispersa. Un punto importante de las transformadas wavelet es la existencia de algoritmos rápidos para su cálculo, habitualmente en forma de banco de filtros.

Sin embargo, no podemos llevar a cabo un análisis MRA estándar en TMs. Al contrario que en una imagen, donde (intuitivamente) podemos interpretar que se utiliza la información de los puntos adyacentes para aproximar el valor de un cierto pixel, la relación espacial entre los nodos de una red es mucho más compleja. Por eso debemos utilizar técnicas específicamente diseñadas para MRA sobre grafos, como la transformada Diffusion Wavelet (DW) [13], que permite no sólo el MRA de grafos (la topología de la red, en nuestro caso) sino también de funciones definidas sobre dichos grafos (la TM, por ejemplo). Los grafos representan la distribución de los nodos en la red subyacente (sobre la cual se enruta la TM) y reflejan la relación espacial natural en la TM. Por ejemplo: dos nodos cercanos en la red es probable que tengan patrones de comportamiento semejantes, como por ejemplo la distribución de su actividad diurna, y es probable que ante alguna anomalía también se vean afectados simultáneamente, ya sea de manera similar u opuesta.

Nuestra contribución es la generalización de la DW a dos dimensiones (necesaria dado que las TM son función del nodo origen y el destino) y su aplicación al modelado de matrices de tráfico. Hemos observado que las matrices de tráfico en el dominio de las DW son dispersas y (relativamente) estables en tiempo, siendo la falta de esa estabilidad un síntoma de anomalías en la TM. Los resultados que presentamos en este documento se han obtenido con datos reales de redes operacionales (GÉANT, Abilene), y muestran que el uso de modelos dispersos es prometedor en las tareas de análisis, síntesis y predicción de matrices de tráfico. Implícitamente estas capacidades hacen que el modelo sea también apropiado para la detección de anomalías, aunque ese no sea el objetivo principal de nuestro trabajo.

El resto del artículo se organiza como sigue. La sección 2 presenta conceptos básicos sobre la transformada wavelet y la Diffusion Wavelet. A continuación de describe el trabajo realizado por otros autores, seguido de la generalización de la DW a 2D. Seguidamente se discuten los resultados obtenidos a partir de datos reales. El artículo finaliza con las conclusiones y las líneas futuras de investigación.

II. WAVELETS Y DIFFUSION WAVELETS

A. La transformada wavelet discreta

En procesado de señal se suelen utilizar métodos basados en wavelets para comprimir y eliminar el ruido de señales o imágenes [14]. La transformada wavelet discreta (*Discrete Wavelet Transform*, DWT) analiza señales unidimensionales (series temporales) a partir de su producto escalar con dos funciones base denominadas función de escalado $\varphi(t)$ y wavelet madre $\psi(t)$, de longitud finita, que son dilatadas (en potencias de 2) y trasladadas para cubrir todo el dominio

temporal de la señal original, obteniendo un análisis de la señal en instantes $t = 2^j$-k (donde j es la escala y k es el desplazamiento temporal). El papel de la función de escalado es el de capturar las bajas frecuencias de la señal, mientras que las altas frecuencias o detalles son analizadas por la wavelet madre. Si las funciones base cumplen ciertas condiciones, la transformada resultante es ortonormal y se puede implementar fácilmente con un banco de filtros paso bajo y paso alto ($h(n)$ y $g(n)$ respectivamente, relacionados con $\varphi(t)$ y $\psi(t)$), como muestra la Fig.2.

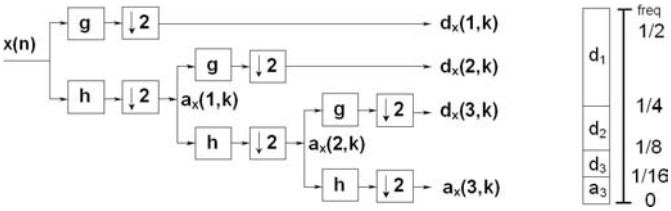


Fig. 2. Izquierda: banco de filtros correspondiente a la DWT para J=3 escalas, obteniendo la aproximación $a_x(3,k)$ y los detalles $d_x(j,k)$ para j=1…3. Derecha: descomposición del espectro normalizado

Los filtros paso bajo nos muestran la aproximación sucesiva en escalas cada vez menos detalladas. Podemos interpretarlo como una imagen que se va difuminando progresivamente, según vaya pasando por más filtros paso bajo. Por otra parte, los filtros paso alto de cada iteración del banco se quedarán con los detalles de alta frecuencia, es decir, la diferencias respecto a la aproximación anterior: $d_x(j,k)$ captura la diferencia entre $a_x(j-1,k)$ y $a_x(j,k)$, donde j es un nivel "inferior" o más difuminado que j-1. Siempre es posible recuperar la señal original invirtiendo el proceso y sintetizando aproximaciones y detalles a partir de los coeficientes de la transformada wavelet.

En términos matemáticos el MRA consiste en obtener un conjunto anidado de subespacios V_j (aproximaciones), $V_1 \supset V_2 \supset \ldots \supset V_j$ y sus complementos ortogonales, los subespacios de los detalles de alta frecuencia $W_j = V_{j-1} - V_j$.

En el dominio frecuencial, la DWT genera una descomposición en subbandas cuyo espectro se divide a la mitad a cada paso. Esto genera un MRA donde la señal original se descompone en una aproximación final a la más baja frecuencia (que tenderá al valor medio de los elementos) y un conjunto de detalles a diferentes altas frecuencias (los coeficientes de la wavelet). Este proceso para generar detalles y aproximaciones se puede generalizar a imágenes 2D, como veremos en la sección IV.

B. *Transformada Diffusion Wavelet*

Las transformadas wavelet mencionadas hasta ahora operan en señales definidas uniformemente en R (series temporales) y R^2 (imágenes). Sin embargo, una TM se define a partir de una red de nodos unidos por enlaces, que se puede representar con un grafo. Las wavelets de difusión [5] (*Diffusion Wavelets,* DW) son una generalización de la transformada wavelet en la que el MRA se puede llevar a cabo en estructuras como variedades (*manifolds*). En topología y geometría diferencial, una variedad es un objeto que generaliza el concepto de curva o superficie a cualquier dimensión [15]. Intuitivamente, es un espacio en la que dos puntos de un cuerpo pueden estar muy próximos en espacio pero alejados en el cuerpo; por ejemplo, en un elipsoide vacío dos puntos enfrentados tienen coordenadas similares, por lo que están cercanos en espacio, pero la distancia que hay que recorrer *a través del cuerpo* para llegar de uno al otro es mayor que la distancia euclidiana (que "saltaría" a través del objeto). Otro ejemplo es el objeto conocido como *swiss roll* o brazo de gitano, una superficie que se enrolla en espiral dentro de sí misma. El camino entre un punto del interior y otro del exterior de la elipse es largo si lo hacemos sobre la superficie, pero mucho más corto si atravesamos el objeto. La clave está en que si definimos funciones sobre la variedad, la noción de vecino o proximidad debe tener en cuenta la topología de la superficie, que localmente se aproxima a R^n, pero globalmente es radicalmente diferente. Un grafo es un caso de variedad.

Fig. 3. Ejemplo de variedad: el *swiss roll*.

Sea el grafo $G\{V, A\}$ (donde V son los vértices y A las aristas). Lo que queremos es aplicar el MRA tanto al grafo como a cualquier función definida en los vértices (por ejemplo, f: $V \rightarrow R$ que asigna un número real a cada vértice del grafo o nodo de la red).

La técnica DW consiste en crear un *operador de difusión* que sea análogo a la función de escalado de la DWT. Aplicar este operador "difumina" la función a estudiar, f, sobre el grafo subyacente. Los valores de la función sobre nodos cercanos (cercanos en la topología del grafo) se mezclarán rápidamente, mientras que los lejanos se mantendrán separados. El uso de este grafo subyacente implica que la DW se adapta intrínsecamente a la topología sobre la que se definen las funciones anteriores.

Matemáticamente, el operador de difusión se representa con la transformada lineal $T f$, donde T se expresa como una matriz aplicada a la función (vector) f. De la misma manera que el abanico de wavelet madre y funciones de escalado es amplio, tendremos varias opciones para elegir T. Un ejemplo simple podría ser una matriz estocástica que representase un paseo aleatorio (*random walk*) definido sobre el grafo. En vez de ello seguimos [16] y elegimos el operador $I - L$, donde I es la matriz identidad y L la matriz Laplaciana normalizada [17] de A, que a su vez es la matriz de adyacencia del grafo. Este grafo está íntimamente relacionado con el *random walk* [16, 17] y tiene los mismos autovalores pero, al contrario que el *random walk*, este operador es simétrico, entre otras características convenientes. En general podemos utilizar cualquier operador T que tenga un autoespectro (espectro de autovalores) decreciente y que esté normalizado de manera que el autovalor más grande sea 1, tal como veremos más adelante.

Una vez definido T, lo *dilatamos* tomando sus potencias. Intuitivamente, lo que hacemos es avanzar la difusión en una unidad de tiempo cada vez que multiplicamos T por sí misma; por tanto, tras n instantes temporales, se aplica la

transformada lineal n veces, es decir $T^n f$. Esto da como resultado un difuminado sucesivo de la función, como se pretendía. En la interpretación del *random walk*, T puede entenderse como la matriz de probabilidades de transición asociadas a una cadena de Markov, y T^n representa la distribución de estos estados tras n instantes temporales, que tenderá a desdibujarse (para una cadena de Markov irreducible) hasta llegar a una distribución equilibrada para $n \to \infty$. Para aplicar el operador dilatado a la función f sólo tenemos que calcular $T^n f$. Análogamente a la transformada wavelet estándar, la DW progresa según potencias de 2, y se toma $T^{2^j} f$. En realidad, y por cuestiones técnicas, lo que en realidad se hace es aplicar T^{2^j} sucesivamente, lo que lleva a la serie T, $TxT^2 = T^3$, $T^3 x T^4 = T^7$, $T^7 x T^8 = T^{15} \ldots$ obteniendo en general $T^{2^j - 1}$ para la escala j.

En grafos, el equivalente natural de descomposición espectral resultante de la DWT es la teoría espectral de grafos: esto es, el estudio de los autovalores y autofunciones de operadores lineales. El Teorema Espectral [17] desemboca en la siguiente representación del operador lineal

$$T = \sum_{i=1} \lambda_i v_i^T v_i \qquad (1)$$

donde λ_i son los autovalores de T y v_i sus autovectores asociados. Si T es un "buen" operador de difusión (con el mayor autovalor normalizado a 1 y con autoespectro decreciente), entonces $|\lambda_i| \le 1$.

Tras el cálculo de cada potencia de T, ignoraremos los autovalores inferiores a un cierto nivel $|\lambda_i| \le \varepsilon$, donde épsilon será un parámetro configurable con un valor pequeño (entre 10^{-3} y 10^{-10} en nuestros experimentos). Unos pocos autovalores (en el caso que quedaran por debajo de ε) se descartan en el primer paso, y el proceso se va repitiendo cuando consideramos las potencias de T.

Los autovalores de T^n son λ_i^n (los autovectores cambian y no siguen ninguna regla predefinida). Cuando $n \to \infty$ todos los autovalores $|\lambda_i| < 1$ tenderán a cero, y finalmente caerán bajo el umbral ε, mientras que el autovalor de valor 1 se mantendrá sin variaciones. De este modo, la aplicación sucesiva de este operador de difusión dividirá el espectro original del grafo en subbandas, y podemos definir el MRA de la siguiente manera: para una escala dada j, los autovectores asociados con $|\lambda_i^{2^j}| \ge \varepsilon$ expanden el subespacio V_j de la aproximación de baja frecuencia, mientras los autovectores asociados con los autovalores descartados en el paso j-ésimo (aquellos que $|\lambda_i^{2^j}| < \varepsilon$ y $|\lambda_i^{2^{j-1}}| \ge \varepsilon$ expanden el subespacio de alta frecuencia o de detalle W_j. A cada paso j, los autovectores que se mantienen se reortonormalizan apropiadamente (con un algoritmo tipo Gram-Schmidt). La Fig. 4 ilustra el proceso. Coiffman y Maggioni [13] presentan un algoritmo rápido para realizar dichos cálculos, y obtener los coeficientes de aproximación y detalle de nivel j (denotados como C_{V_j} y C_{W_j} respectivamente) proyectando la función sobre V_j y W_j.

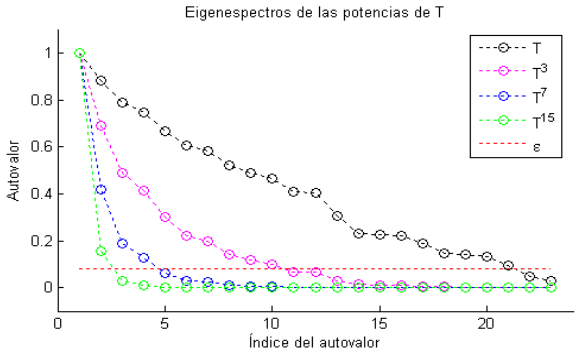


Fig. 4. Autoespectro de las potencias de un operador T. Los autovalores pro debajo de ε en cada paso generan los subespacios W_j y los restantes, los V_j.

III. TRABAJOS PREVIOS

Una red IP se puede interpretar como un grafo, cuyos vértices representan routers o puntos de presencia (*Points of Presence*, PoPs) y cuyas aristas corresponden a los enlaces entre éstos. En este contexto definimos la matriz de tráfico como la matriz que describe los volúmenes de tráfico que atraviesan una red entre los nodos de entrada y salida de la misma, medidos en un tiempo de observación determinado. El primer problema con el que nos encontramos es cómo obtener estas medidas del tráfico en los enlaces.

La medida directa mediante software que corre en el router, como *Cisco NetFlow* o similares es la solución más directa. Sin embargo, es una aplicación que no está disponible en todos los routers y consume mucha CPU. En el peor de los casos esta aplicación puede afectar al trabajo del router, llegando al punto en que la medición de estos datos afectase a la capacidad del propio dispositivo para enrutarlos.

La alternativa es el uso de los contadores de las MIB de SNMP presentes en las interfaces de red de los routers. Su gran ventaja es que está presente en todos los routers y es fácil de consultar. Además, prácticamente no consume recursos de CPU. Sin embargo, sólo proporciona la cantidad de bytes transmitidos a través del enlace, y no el tráfico enviado desde cada nodo a cualquier otro; es decir, sabremos que por un enlace entre A y B hay cierto tráfico, pero desconoceremos el tráfico que inyecta A y el que sale en B ya que se mezcla con el tráfico de otros flujos (C→D) que viajan por ese mismo enlace.

Sin embargo, podemos atacar el problema mediante una medida indirecta o inferencia. Las medidas SNMP de los enlaces, **y,** se relacionan con la TM, **x,** según la expresión

$$\mathbf{y} = A\mathbf{x} \qquad (2)$$

donde A es la matriz de enrutamiento[1] y **x** es la matriz de tráfico expresada como vector. Desgraciadamente las dimensiones de **y** son mucho menores que las de **x**, lo que provoca que problema de deducir una TM a partir de los datos SNMP se convierta en un problema inverso sin solución, en el que necesita información adicional,

[1] A tiene N^2 columnas (correspondientes a cada una de las NxN rutas posibles entre los N nodos de origen/destino) y K filas (correspondientes a cada uno de los enlaces de la red). Las entradas de $A(i,j)$ son 1 si el enlace i está en la ruta j, y 0 en otro caso.

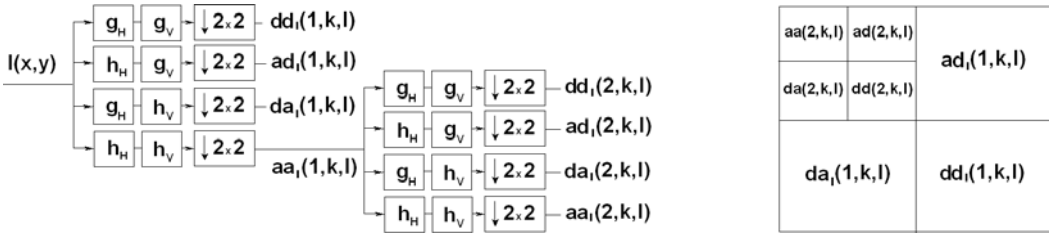


Fig. 5: Banco de filtros asociado con la transformada wavelet 2D (izquierda) y descomposición espectral correspondiente (derecha).

habitualmente en forma de un modelo a priori de las entradas de la TM. Ejemplos de esta información adicional en el contexto de Internet son modelos de Gauss [4], logit-choice [5], de Poisson [6], o de gravedad [7].

Otros estudios sobre el modelado de TMs se han aplicado con éxito en la detección de anomalías [18]. Esos documentos usan PCA (Principal Component Analysis) sobre las TMs como series temporales, y se centran en la correlación de los elementos de las TMs para separar los componentes periódicos del tráfico de las fluctuaciones aleatorias y otras anomalías. Sin embargo, no queda claro cómo las estructuras descritas en [18] llevarían a un modelo simple para usar en la síntesis de matrices de tráfico.

Por otra parte, el modelo de gravedad [20] parece ser un buen paso en el camino hacia la obtención de un modelo con significado físico. A lo largo del siglo XX se han definido modelos basados en la ley de la gravedad de Newton para explicar, por ejemplo, flujos de transporte por carretera, volúmenes de comercio o movimientos migratorios. En el modelado de tráfico se sigue el mismo principio: así como la fuerza entre dos cuerpos viene determinada por la masa de los mismos, la distancia entre ellos y una constante, el modelo de gravedad asume que el tráfico entre dos nodos A y B será proporcional al producto del tráfico total que sale de A y del tráfico total recibido por B. Dos nodos "troncales", de los que dependan muchos nodos menores, verán más tráfico entre ellos que el que verían dos nodos más pequeños; esto es, si hay más "población" detrás de un nodo, mayor volumen de tráfico emitirá o recibirá dicho nodo. El modelo se reduce a *2N* parámetros, que corresponden a las *N* distribuciones marginales de probabilidad del tráfico emitido y las *N* del tráfico recibido, una de cada tipo para cada nodo.

Sin embargo se ha comprobado que el modelo de gravedad no es *per se* un modelo completo, ya que en realidad es un modelo de rango 1 (solo hay una fila o una columna linealmente independientes en la TM sintética generada como producto de las *NxN* probabilidades) y reduce demasiado la dimensionalidad del problema como para ser útil. Sin embargo, es un excelente inicio para encontrar el punto más cercano que pertenece al espacio de soluciones de la ecuación (1); es decir, la solución de $\mathbf{y} = A\mathbf{x}$ más cercana al modelo de gravedad.

Una primera aproximación a la aplicación de MRA a grafos fue llevada a cabo por Crovella y Kolaczyk en [19], donde las Graph Wavelets (GW) se introdujeron como una extensión a la transformada wavelet 2D. GW permite computar la diferencia de carga entre enlaces separados por un número determinado de nodos; aquí el concepto de escala propio de las wavelets se sustituye por el de "distancia en saltos entre enlaces". Los autores también muestran cómo esa herramienta se puede utilizar también para detección de anomalías. Sin embargo, no dispone de un algoritmo computacional rápido, y los resultados de la transformada no son ortonormales. Finalmente, la transformada Graph Wavelets no representa el tráfico de manera dispersa, sino que resulta en una descomposición redundante muy similar a la que obtenemos con la transformada wavelet continua.

La transformada Diffusion Wavelets ofrece una base matemática sólida (de la que carecen las Graph Wavelets) al análisis MRA sobre grafos. DW se puede entender como una generalización de las wavelets en grafos, permitiendo más libertad a la hora de escoger una función wavelet madre (la función nuclear o básica subyacente), añadiéndole las distancias generalizadas en el grafo, una base ortonormal y un algoritmo computacional rápido. Que nosotros conozcamos, la única aplicación relevante de las wavelets de difusión en el contexto de las redes de ordenadores es [20], donde Coates *et al*. abordan el problema de encontrar el mínimo número de medidas necesarias para monitorizar extremo a extremo ciertas métricas de una ruta (por ejemplo el retardo en la capa IP o la tasa de error de bit en la capa física de una red óptica). Los autores construyen un operador de difusión en un grafo alternativo complementario donde los nodos son las rutas de la red original y los enlaces son una medida de similitud entre las rutas (la proporción de enlaces compartidos, según los datos de enrutamiento) y aplican la transformada DW. La dispersión implícita de los datos analizados en el dominio de la transformada, junto con el uso de técnicas de inferencia por dispersión, permite una excelente monitorización con un número reducido de dispositivos (por ejemplo, la media del retardo extremo a extremo para todas las rutas de la red se puede medir con alta precisión monitorizando tan solo el 7% de las rutas).

IV. Transformada DW en 2D

Las matrices de tráfico se pueden representar como funciones bidimensionales $F(v_1, v_2)$ de pares de vértices donde v_1 es el nodo origen del tráfico, v_2 el nodo de destino, y $F(v_1, v_2)$ es el volumen de tráfico entre v_1 y v_2. Dado que la transformada Diffusion Wavelet en su forma original sólo se puede aplicar a funciones bidimensionales, debemos extender y adaptar las DWs al caso 2D.

Las wavelets clásicas unidimensionales se pueden utilizar en análisis de imagen para construir una base en 2D combinando la aplicación de sucesivos filtros paso alto y paso bajo en las dimensiones horizontal y vertical de la imagen de entrada $I(x, y)$, generando así 4 subbandas en cada escala con las cuatro combinaciones posibles de esos dos filtros: PB-PB (aproximación) y PB-PA, PA-PB y PA-PA (detalles). Tras diezmar las salidas adecuadamente, se itera el proceso tomando como entrada la aproximación (la salida del primer PB-PB), como muestra la Fig. 5. En líneas generales es el mismo proceso que en la estructura ilustrada

en la Fig. 2 donde, de la misma manera, tomábamos aproximación y detalles en cada iteración, solo que ahora tendremos 3 subbandas para los detalles y una para las aproximaciones.

Análogamente, la versión 2D de la DW transforma la función $F(v_1, v_2)$ proyectándola una vez sobre cada subespacio determinado por el operador de difusión de la DW clásica en 1D (el de detalles y el de aproximación). Los detalles del algoritmo varían, pero intuitivamente el proceso es similar. De manera similar a como hacíamos con las DW en 1D, llamaremos, en este caso, C_{VV_j}, C_{VW_j}, C_{WV_j} y C_{WW_j} a los coeficientes de la transformada que corresponden a los subespacios VV_j, VW_j, WV_j y WW_j respectivamente (donde *VV* se refiere al subespacio de la aproximación a baja frecuencia y *WW* al subespacio del detalle a más alta frecuencia). La transformada 2D resultante verifica las propiedades deseables de ortonormalidad, invertibilidad (podemos reconstruir la función original a partir de sus coeficientes) y separabilidad.

La Fig. 6 presenta un ejemplo ilustrativo de este proceso de separación en subbandas respecto a un grafo inicial con 10 nodos. La transformada DW en 1D descompone el espectro de este grafo en 3 subbandas con autovalores 2, 3 y 5 para las subbandas W_1, W_2 y V_2 respectivamente, mientras que la transformada en 2D lo divide en la aproximación VV_2 y los detalles VW_2, WV_2, WW_2, VW_1, WV_1 y WW_1, incluyendo cada uno un conjunto de $n \times m$ autovalores/autovectores.

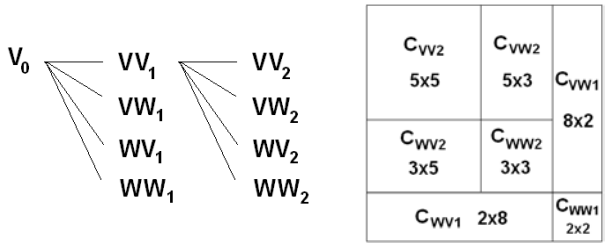


Fig. 6. Ejemplo de una descomposición generada por la DW 2D para J=2.

V. Análisis Multi-Resolución de Matrices de Tráfico

En nuestros primeros experimentos con la herramienta DW en 2D hemos estudiado más de 20000 matrices de tráfico pertenecientes a dos conjuntos de datos de las redes de Abilene y GÉANT, con 12 y 23 PoPs respectivamente. La granularidad de las TM es de 5 minutos en el caso de las de Abilene, y de 15 minutos en las de GÉANT. Para más detalles sobre estos datasets véase [11, 21, 22].

Estas TMs fueron analizadas con la wavelet de difusión 2D con dos objetivos: visualizar como el proceso de difusión afectaba a una matriz de tráfico para desarrollar nuestra intuición sobre la descomposición multi-resolución (y comprobar la invertibilidad y propiedad de reconstrucción), y evaluar la compresibilidad obtenida con el método 2D. El primer operador utilizado es el random walk (normalizado al estilo *I – L* tal como se describe en la Sección II.b), en la que la probabilidad de salir de un nodo a través de un cierto enlace es el inverso de la cantidad de enlaces conectados al nodo (el grado del vértice, en nomenclatura de teoría de grafos). Nótese que en principio este operador no captura necesariamente el encaminamiento de los flujos sobre la red ni otras propiedades que pueda tener la distribución del tráfico sobre la red, pero es el ejemplo más simple.

La Fig. 8 muestra los resultados obtenidos con la transformada DW de una TM representativa de Abilene (2 de marzo de 2004, de 12:00 a 12:05). Las imágenes muestran reconstrucciones de TMs a partir de sus coeficientes de aproximación (C_{VV_j}) y de detalle (C_{VW_j}, C_{WV_j} y C_{WW_j}) a cada escala *j*. El operador de difusión analizado ha sido el random walk sin ponderar (peso 1 en los enlaces) y normalizado, y la precisión ha sido de $\varepsilon = 10^{-7}$. Como no hay autovalores descartados en los dos primeros subespacios de detalle, la reconstrucción de las aproximaciones asociadas son idénticas a la TM original, y no se muestran en la figura. Se puede ver claramente el efecto de las bajas frecuencias al aplicar repetidamente el operador de difusión, sucesivamente, en las aproximaciones, junto con los componentes de alta frecuencia en los detalles. Nótese que se está preservando la topología de red que aparece en la Fig. 1, en la que, por poner un ejemplo fácilmente visible, el nodo 12 (ATLA-M5) cuelga únicamente del nodo 8 (ATLA), y se puede observar que la difusión llega a ATLA-M5 mucho más tarde (en una potencia más grande, en un tiempo más tardío, en una escala más elevada) que en el núcleo de la red.

En cuanto a la compresibilidad de la TM, hemos llevado a cabo varias pruebas tomando las matrices de ambos conjuntos de datos en quincenas o meses enteros, para averiguar cuánta energía de las matrices originales se mantiene en los primeros coeficientes. La Tabla 1 muestra los resultados para dos conjuntos mensuales de matrices representativos. Los resultados obtenidos con otros conjuntos mensuales o quincenales son consecuentes con los que se muestran en la Tabla 1, y confirman la dispersión de la representación DW: en promedio, el 15% de los coeficientes contiene el 90% de la energía original de la TM. La Figura 7 complementa los resultados del estudio sobre la compresibilidad mostrando la gráfica del error medio cuadrático (*Mean Square Error*, MSE), de las TMs reconstruidas respecto el porcentaje de coeficientes usados en esta reconstrucción. Cabe destacar que pese a las diferencias entre Abilene y GÉANT, los resultados son prácticamente idénticos, ilustrando cómo la DW con el operador más simple es, sin embargo, capaz de capturar la estructura de la matriz de tráfico (y decorrelarla, en cierto modo), independientemente de la topología de la red.

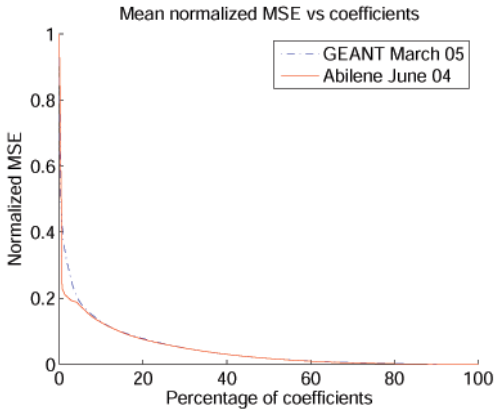


Fig. 7. Error cuadrático medio normalizado en función del porcentaje de coeficientes de la DW, para dos trazas de TMs de un mes de duración.

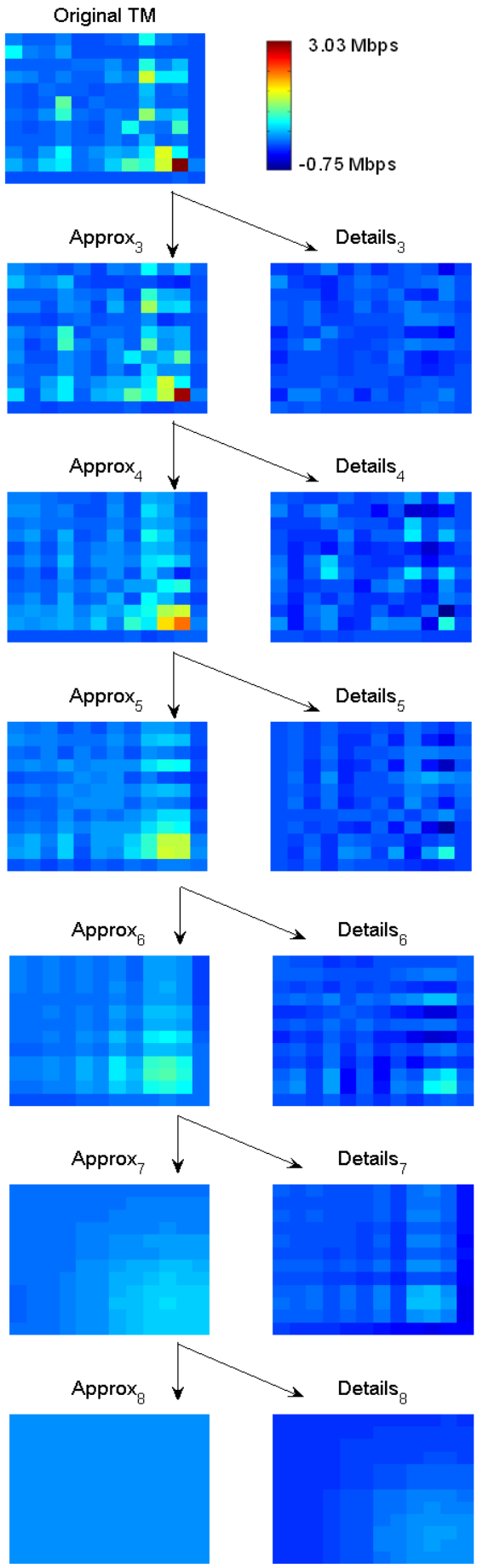


Fig. 8: Aproximaciones y detalles del análisis multi-resolución de la matriz de tráfico TM-2004-03-02-1200 de Abilene.

	Energía preservada	
Traza en estudio	80%	90%
Abilene, Junio 04	4 coefs. (2.8%)	21 coefs. (14.6%)
GÉANT, Marzo 05	24 coefs. (4.5%)	75 coefs. (14.2%)

Tabla 1. Porcentaje de coeficientes necesarios para preservar ciertas fracciones de la energía original en la matriz de tráfico.

Finalmente, hemos observado que hay una característica propia de cada dataset (a modo de firma), relacionada con el orden de los coeficientes (ordenados según su contribución a la energía total de la TM), que además es consistente temporalmente, excepto cuando hay alguna anomalía en la serie de TMs (como por ejemplo un cambio drástico del volumen de tráfico). La Fig. 9 muestra un ejemplo de esta característica: en la figura se observa el orden relativo de los 20 coeficientes mayores de una serie de 14 días en marzo de 2004 para el caso de Abilene. Se puede ver un cambio claro alrededor de la TM número 3500 (primeras horas del 14 de marzo), que coinciden en el tiempo con un cambio estructural notable en las matrices de tráfico originales (un aumento sustancial del tráfico dirigido a Houston desde varias fuentes). Esto sugiere que los cambios estructurales en las TMs pueden ser detectados monitorizando tan sólo un pequeño número de coeficientes (en vez de las 144 rutas posibles) siempre y cuando la energía que contengan sea significativa de la matriz de tráfico original. Esto está aún lejos de convertirse en un algoritmo de detección de anomalías basado en MRA, pero la transformada DW parece potencialmente útil para esta labor.

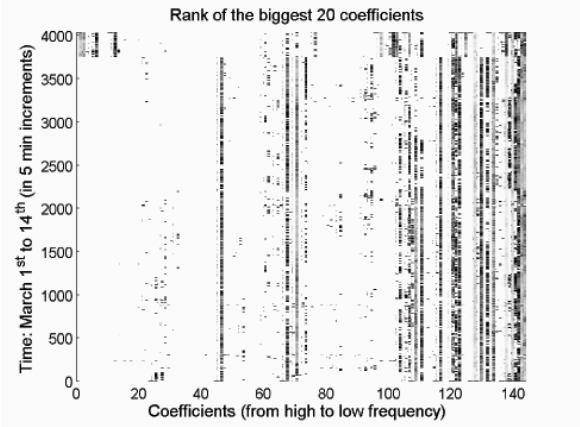


Fig. 9. Firma característica de la traza Abilene de Marzo 2004.

Como hemos visto, la elección del operador de difusión es un factor importante en el proceso, ya que define la base sobre la que se proyecta la función bajo estudio y determina en gran medida los resultados esperados. Los resultados presentados para el operador más sencillo (el random walk sin pesos) son prometedores, pero queremos ir más allá, y en esto se está centrando nuestra labor de investigación. A modo de ejemplo, presentamos a continuación un operador de difusión basado en el modelo de gravedad. La idea es que en vez de trabajar sobre el grafo correspondiente a la topología, nos abstraemos de la red subyacente y definimos un grafo completo en el que los enlaces entre nodos tienen el peso correspondiente al modelo de gravedad; es decir, hacemos más "cercanos" los pares de nodos que intercambian más tráfico y más "lejanos" los que

intercambian menos tráfico. Nótese que el modelo de gravedad puede obtenerse fácilmente a partir de los contadores SNMP y no necesita de las medidas completas tipo Netflow para deducirlo en una red real. Por el contrario, una posible crítica a este operador es que es dependiente de los propios datos de tráfico que queremos modelar, pero en realidad lo que estamos utilizando es una aproximación de rango 1 y no todos los datos de la TM, como hemos comentado anteriormente, y en todo caso son datos disponibles directamente de las medidas SNMP. Por otro lado, los coeficientes del modelo de gravedad son relativamente estables con el tiempo (en cierto grado, son un invariante de la TM [7]), y se justifica que esta invarianza se utilice en el modelo. Los resultados obtenidos respecto a la compresibilidad (*sparsity*) del modelo mejoran notablemente, tal como se aprecia en la Fig. 10.

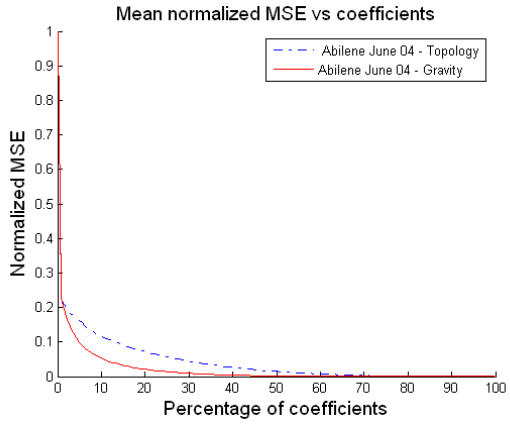


Fig. 10. Error cuadrático medio normalizado en función del porcentaje de coeficientes de la DW para dos trazas Abilene con dos operadores diferentes.

VI. Conclusiones

El presente artículo presenta el uso de las wavelets de difusión en la aproximación dispersa de las matrices de tráfico, siendo el objetivo a largo plazo el de encontrar modelos para dichas matrices. Sin embargo, queda por realizar mucho más trabajo en cada uno de los siguientes campos: (i) transformar la compresibilidad de la matriz de tráfico en un modelo físico; (ii) usar ese modelo para resolver los diversos problemas relacionados con las TM: inferencia de TMs (explotando la reducción dimensional), síntesis y predicción de TMs, y detección de anomalías; (iii) investigar cómo el modelo disperso DW se relaciona con otros resultados como pueden ser el modelo de gravedad o los basados en PCA; (iv) explorar otros operadores de difusión, especialmente los que son capaces de introducir la matriz de encaminamiento en el operador; (v) investigar cómo las topologías de red se pueden representar y estudiar desde el enfoque del MRA y (vi) desarrollar el modelo de gravedad como punto de partida para diseñar un operador de difusión más adecuado para el análisis de las matrices de tráfico.

Agradecimientos

Este trabajo se realizó parcialmente mientras David Rincón visitó la University of Adelaide (Australia) y IPAM-UCLA (Los Ángeles, USA), y ha sido financiado parcialmente por la UPC (Mobilitat 2007/08), el Ministerio de Ciencia (TSI-2005-06092) y por la red EuroNF (INFSO-ICT-216366).

Los autores quieren expresar su agradecimiento a Yin Zhang de la University of Texas por los datos de Abilene, y a Steve Uhlig y el equipo del proyecto TOTEM por los datos de GÉANT, así como a Matthew Roughan (University of Adelaide), Walter Willinger (AT&T) por sus comentarios e ideas, y a Mauro Maggioni por publicar el código Matlab de la Diffusion Wavelet y por sus comentarios. Los autores también quieren agradecer a los revisores anónimos sus sugerencias y comentarios.

Referencias

[1] D.L. Alderson, H. Chang, M. Roughan, S.Uhlig, and W. Willinger. The many facets of Internet topology and traffic. *Networks and Heterogeneus Media,* 1(4): 569-600, Dec. 2006.

[2] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True. Deriving traffic demands for operational IP networks: Methodology and experience. *IEEE/ACM Transactions on Networking*, pp. 265-279, Jun. 2001.

[3] M. Roughan, M. Thorup, and Y. Zhang. Traffic engineering with estimated traffic matrices. *Procs. of ACM IMC 03*, pp. 248-258, 2003.

[4] J. Cao, D. Davis, S. V. Wiel, and B. Yu. Time-varying network tomography. *Journal of the American Statistical Association*, 95(452):1063-1075, 2000.

[5] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic matrix estimation: existing techniques and new directions. *SIGCOMM Comput. Commun. Rev.*, 32(4):161-174, 2002.

[6] Y. Vardi. Network tomography: estimating source-destination traffic intensities from link data. *J. of the Am. Stat. Ass.*, 91:365-377, 1996.

[7] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale IP traffic matrices from link loads. *ACM SIGMETRICS*, pp. 206-217, San Diego, California, Jun. 2003.

[8] Y. Zhang, M. Roughan, C. Lund, and D. Donoho. An information-theoretic approach to traffic matrix estimation. *ACM SIGCOMM*, pp. 301-312, Aug. 2003.

[9] B. Fortz, J. Rexford, and M. Thorup. Traffic engineering with traditional IP routing protocols. *IEEE Comm. Magazine*, 40(10):118-124, Oct. 2002.

[10] A. Nucci, A. Sridharan, and N. Taft. The problem of synthetically generating IP traffic matrices: Initial recommendations. *SIGCOMM Comput. Commun.Rev.*, 35(3), 2005.

[11] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon. Providing public intradomain traffic matrices to the research community. *SIGCOMM Computer Communication Review*, 36(1):83-86, 2006.

[12] M.D. Adams, The JPEG-2000 still image compression standard, ISO/IEC JTC 1/SC 29/WG 1 N 2412, ISO/IEC, 2001.

[13] R. R. Coifman and M. Maggioni. Diffusion Wavelets. *Applied and Computational Harmonic Analysis*, 21(1):53-94, Jul. 2006.

[14] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[15] Wikipedia. Artículo Variedad (matemática) en español [online] http://es.wikipedia.org/wiki/Variedad_(matemática)

[16] S. Mahadevan and M. Maggioni. Proto-value Functions: A Laplacian Framework for Learning Representation and Control in Markov Decision Processes. *The Journal of Machine Learning Research*, 8:2169-2231, 2007.

[17] F. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, Feb. 1997.

[18] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft. Structural analysis of network traffic flows. *SIGMETRICS Perf. Eval. Rev.*, 32(1):61-72, 2004.

[19] M. Crovella and E. Kolaczyk. Graph wavelets for spatial traffic analysis. *Proceedings of IEEE Infocom*, pp. 1848-1857, Apr. 2003.

[20] M. Coates, Y. Pointurier, and M. Rabbat. Compressed network monitoring for IP and all-optical networks. *Proceedings of ACM IMC 2007*, pp. 241-252, 2007.

[21] Y. Zhang's homepage. Abilene Traffic Matrices [online]. http://www.cs.utexas.edu/yzhang/research/AbileneTM/

[22] TOTEM project. http://totem.run.montefiore.ulg.ac.be/datatools.html.

[23] D. Rincón, M. Roughan, W. Willinger. Towards a Meaningful MRA of Traffic Matrices. *Proceedings of ACM IMC 2008*, pp. 331-336, 2008.

Algoritmo Distribuido para la Asignación Dinámica de Recursos en Redes EPON

Marilet De Andrade, Paola Garfias
Sebastià Sallent, Lluis Gutiérrez
Departamento de Ingeniería Telemática
Universidad Politécnica de Cataluña
Av. Canal Olímpic 15, Castelldefels, España
marilet@entel.upc.edu, paola.garfias@entel.upc.edu
sallent@entel.upc.edu, lluis.gutierrez@entel.upc.edu

Anny Martínez, Pedro Vizarreta
Dayana Sánchez, Mónica Huerta
Departamento de Electrónica y Circuitos
Universidad Simón Bolívar
Valle de Sartenejas, Caracas, Venezuela
angama168@gmail.com, pedrovi86@gmail.com
sanchez.dayana@gmail.com, mhuerta@usb.ve

***Resumen*- Las redes de acceso del futuro requieren cada vez más capacidad, la red EPON basada en Ethernet destaca por sus ventajosas prestaciones entre las múltiples soluciones propuestas a corto y medio plazo. Uno de los aspectos clave en este tipo de redes es la asignación dinámica del ancho de banda a los usuarios que en el estándar de la red EPON (IEEE 802.3h) se ha dejado abierto para permitir implementar el mecanismo más eficiente. En este artículo se propone y analiza exhaustivamente un algoritmo dinámico y distribuido para la asignación de recursos, en el que los dispositivos de usuario intervienen activamente en la ejecución de tal tarea. Para este trabajo se ha construido un entorno de simulación fiel a las especificaciones del estándar utilizando la herramienta OPNET Modeler. A través de los resultados de las simulaciones se puede ver que este esquema presenta significativas mejoras en sus prestaciones si se compara con el algoritmo centralizado de referencia en la literatura, especialmente cuando la carga de la red es elevada. Además, el algoritmo propuesto se comporta de forma estable y más eficiente cuando variamos la distancia entre los terminales de usuario y la cabecera de la red de acceso.**

***Palabras Clave*- Redes Ópticas Pasivas, EPON, distribución dinámica de ancho de banda, algoritmo distribuido**

I. INTRODUCCIÓN

En los últimos años el incremento drástico de la capacidad de las redes de transporte, aunado al aumento exponencial de las aplicaciones y servicios que demanda usuario residencial y la pequeña y mediana empresa ha creado un problema de cuello de botella en la red de acceso. Sin lugar a duda, la infraestructura necesaria se basa en las redes ópticas pasivas dadas las altas capacidades que puede proveer. En particular las redes ópticas pasivas (PON) se consideran como las redes de acceso más prometedoras gracias a su bajo costo de despliegue. Una PON es una red punto-multipunto formada básicamente por fibra óptica y uno o varios divisores ópticos pasivos. Con ello se ahorra en costos de mantenimiento, equipamiento de distribución remota, suministro de energía e infraestructura de fibra óptica [1].

Actualmente existen dos ramas de estandarización de las PON de acuerdo a la tecnología de capa 2 que se utilice: ITU-T e IEEE. La primera recoge la PON basada en ATM tales como APON y BPON (G.983.x) y la basada en GFP (*Generic Framing Protocol*) conocida como GPON (G.984.x). El estándar de la IEEE 802.3ah [2] se basa en Ethernet (EPON) y es la tecnología más prometedora del mercado. Actualmente su despliegue es muy amplio especialmente en el continente asiático.

Una EPON es una red óptica pasiva que encapsula todos los datos en tramas Ethernet. Tal como se especifica en el estándar, se tienen dos canales definidos por dos longitudes de onda distintas: el canal ascendente (desde el nodo de usuario hacia el nodo de red) y el canal descendente (desde el nodo de red hacia el nodo de usuario). El canal descendente lo utiliza el nodo de red o OLT (*Optical Line Terminal*) para difundir el tráfico hacia todos los nodos de usuario de modo que la información la reciben todos en modo broadcast y es la dirección MAC la que permite al terminal determinar cual es la información dirigida a él. El canal ascendente es compartido por todos los nodos de usuario en tiempo y requiere un control de acceso al medio para evitar las colisiones de las transmisiones de los dispositivos de usuario u ONU (*Optical Network Unit*) ya que las colisiones no están permitidas según lo señala el estándar. Para ello se especifica el uso del protocolo MPCP (*Multi-Point Control Protocol*) que permite el arbitraje del acceso al medio mediante el intercambio de mensajes de control, en particular el mensaje para que el ONU pueda solicitar el ancho de banda que necesita (mensaje Report), y el mensaje para que el OLT asigne la ventana de transmisión a cada usuario, es decir, el comienzo y la duración de la transmisión del ONU (mensaje Gate). El método a seguir para coordinar las diferentes ventanas de transmisión de cada ONU en el canal ascendente, así como la gestión de los recursos disponibles de acuerdo a ciertos requerimientos de calidad de servicio no entran dentro de las especificaciones del estándar y se dejan libres para su implementación. Por ello han sido numerosos los trabajos de investigación que se centran en el estudio de algoritmos de distribución dinámica del ancho de banda o DBA (*Dynamic Bandwidth Allocation*) en los últimos años.

En este artículo presentamos el estudio de un algoritmo DBA de tipo distribuido propuesto previamente por nuestro grupo. En este artículo pretendemos mostrar las ventajas de este esquema comparado con un esquema de referencia denominado IPACT [3]. Se presenta aquí también el desarrollo de modelos estándar sobre la plataforma de simulación *OPNET Modeler* [4]. En particular se estudia la variación de las prestaciones del algoritmo en función de la distancia entre los terminales de usuario y la cabecera central de la red EPON.

El manuscrito está organizado de la forma siguiente: en la sección II se hace un análisis de los trabajos previos que se han realizado en relación al esquema que se propone; en la

sección III se presenta formalmente el algoritmo distribuido propuesto; la sección IV describe la construcción de un entorno de simulación, que cumple con el estándar EPON y se basa en el simulador *OPNET Modeler*; en la sección V se presentan y discuten los resultados obtenidos, destacando las ventajas obtenidas con nuestra propuesta; y finalmente la última sección está dedicada a comentar las conclusiones del trabajo.

II. Algoritmos de Asignación Dinámica de Recursos: Trabajos Previos

Los algoritmos DBA que se han propuesto son en su mayoría centralizados, y entre ellos existen dos tendencias. Una de ellas propone la asignación de ranuras de tiempo de tamaño fijo, lo que se conoce como TDMA fijo [5], donde se podría garantizar ancho de banda y retardos constantes, pero generando un elevado nivel de sub-utilización en el canal ascendente. Otras contribuciones aportan variaciones más elaboradas [6] y [7]. Se obtienen importantes mejoras en cuanto a la eficiencia en el uso del canal pero no elimina del todo la subutilización del canal.

Mientras que la segunda tendencia propone la asignación de ranuras de tiempo variables de un tamaño acorde a las solicitudes de los usuarios, pero limitando el máximo que puede transmitir cada usuario, como se propone en [3]. Variaciones de este trabajo consideran el uso de diversos planificadores, tales como [8] y [9], entre otros. Sin embargo, estos esquemas suelen tener menos control sobre el estado de la red para realizar una correcta planificación. Por otro lado, existe un problema adicional derivado de los retardos añadidos relativos a la espera por parte del OLT para recibir la mayor cantidad de mensajes REPORT a fin de poder realizar la planificación de recursos a distribuir.

Podemos concluir que los mecanismos mencionados son del tipo centralizado, es decir, el algoritmo DBA planificador del ancho de banda que se asigna a cada usuario reside y se ejecuta en el OLT. También existen algunos trabajos en los que se investiga mecanismos descentralizados, donde son los ONUs quienes determinan sus respectivas ventanas de transmisión. En la mayoría de esquemas descentralizados es necesario cambiar la arquitectura de la red tal que se pueda contar con un canal de retorno que permita a los ONUs escuchar el canal y conocer las transmisiones que se han efectuado. En muchos casos es necesario extender una fibra adicional entre cada ONU y el splitter o divisor óptico. Además el splitter de la PON convencional se convertiría en un arreglo de dos splitters que permiten realimentar la señal hacia la fibra de retorno.

La primera pregunta que aparece cuando se plantea el uso de Ethernet en la red PON, especialmente el uso del canal ascendente de forma distribuida, es ¿por qué no implementar el protocolo MAC más conocido: CSMA/CD? En [5] se explica que el mecanismo de acceso al medio por contención (CSMA/CD) es muy difícil de implementar en la EPON dado que, debido a las características direccionales del splitter óptico, los ONUs no pueden detectar una colisión en OLT. Aunque el OLT puede informar a los respectivos ONUs de la ocurrencia de una colisión, esto implicaría mayores retardos de propagación y la EPON reduciría enormemente su eficiencia.

En el trabajo de Chae et al. [10] se realizan experimentos sobre una red de tipo estrella EPON con el algoritmo de CSMA/CD llegando a obtener una elevada eficiencia en el uso del medio. Estos resultados se basan en el re-direccionamiento óptico para que los ONUs puedan detectar las colisiones, es decir, los ONUs también escuchan el canal por el que transmiten. Para resolver el problema expuesto en [5], aquí se propone no incluir al OLT en el proceso de detección de colisiones, solo se aplica CSMA/CD entre los ONUs. Para distancias de distribución (de splitter a ONU) del orden de los 100 metros, la eficiencia es el 99%. Aunque es un resultado interesante, en la práctica esta distancia oscila entre 1 y 2 Km. como mínimo. Adicionalmente, y en general para los mecanismos descentralizados, se requiere un cambio de estructura diferente a la expuesta por el estándar. Por otra parte, este mecanismo no es determinístico y resulta incompatible cuando se pretende ofrecer servicios garantizados.

En [11], Y. Tse et al. proponen un esquema basado en CSMA/CA, en el que existe un canal adicional que se usa para la emisión de tonos antes y durante la transmisión de datos de cada ONU (los datos se transmiten por otro canal distinto al de tonos). Como cada ONU posee un tono distinto, un ONU puede saber si otro ONU está transmitiendo o intentando hacerlo al mismo tiempo. De esta forma se evitan las colisiones, con la ventaja de no necesitar mensajes de control. Este algoritmo presenta las mismas desventajas que CSMA/CD, ya que no es un mecanismo determinístico capaz de soportar calidad de servicio.

Una alternativa es el protocolo Full-RCMA presentado por Fuh et al. [12]. Este mecanismo necesita un cambio en la estructura física de la red EPON. En el splitter se incorpora un acoplador óptico de retorno que se envía a los ONUs a través de una fibra adicional entre el splitter y cada ONU. De esta forma los ONUs pueden conocer lo que se ha transmitido al medio y saber si su transmisión ha colisionado con otra o no. El protocolo Full-RCMA posee dos partes dentro de un ciclo que se repite periódicamente: el de solicitudes y el de datos. En el primero existe contención por el uso del medio para que cada ONU pueda solicitar permiso de transmisión al OLT. Uno de los ONUs se designa como "ganador" y es quien se encarga de ordenar los ONUs cuyas solicitudes se hayan efectuado exitosamente. El OLT no interviene en el proceso de administración del ancho de banda. Los ONUs entre sí administran los tiempos de transmisión sin que ocurran colisiones de los datos. Este algoritmo implica un cambio explícito en el protocolo MPCP especificado por el estándar y en la estructura física de la PON.

S. Sherif et al. [13] proponen otro esquema descentralizado que utiliza el protocolo MPCP como base. Básicamente se divide el ciclo en tres partes: control, espera y datos. En la parte de control todos los ONUs (mediante el canal de retorno) reciben todos los mensajes REPORT. En intervalo de espera cada ONU ejecuta el algoritmo DBA y finalmente, en la fase de datos, los ONUs transmiten los datos sin colisiones. Lo más significativo de este esquema es que es compatible con el estándar en cuanto al uso del protocolo MPCP. Por otra parte, el tiempo de espera tiene un efecto negativo aunque no muy significativo sobre el retardo.

En este artículo se estudia un mecanismo descentralizado que no requiere cambios en la estructura física de la PON. Por otro lado los mensajes de control deben transportar información extra relativa al estado de la red. La ventaja de utilizar un mecanismo de planificación distribuída es que los

dispositivos de usuario deciden la cantidad a transmitir de forma dinámica considerando el estado general de la red, el cual reciben a través de los mensajes Gate provenientes del OLT. A continuación se describe el algoritmo propuesto.

III. Algoritmo Distribuido para la Asignación Dinámica de Recursos en las Redes EPON

El planificador DBA que presentamos a continuación es de tipo descentralizado y además no requiere cambios en la estructura física de la PON. La única modificación necesaria en esta propuesta es que los mensajes de control deben transportar información extra relativa al estado de la red. La ventaja de utilizar un mecanismo de planificación distribuída es que los dispositivos de usuario deciden la cantidad a transmitir de forma dinámica considerando el estado general de la red, el cual reciben a través de los mensajes GATE provenientes del OLT en forma de un vector de pesos, o sea, la proporción instantánea de ancho de banda utilizado, y es entonces cuando el dispositivo de usuario (ONU) calcula el ancho de banda a transmitir. Este cálculo lo realiza la ONU teniendo en cuenta el tamaño actual de la cola. La información que necesita el ONU para realizar este cálculo se recibe a través del mensaje de control Gate y para ellos proponemos utilizar el campo reservado "PAD/Reserve" de la cabecera el cual está disponible. Cada ONU enviará al OLT un parámetro extra que representa su propio peso dentro del mensaje de control Report. Este método ha sido propuesto anteriormente de forma preliminar por los autores de este artículo en [14] y recibe el nombre de DDSPON (*Dynamic Distributed Scheduler for ePON*).

En un sistema PON se tienen N ONUs, y cada ONU i posee un peso nominal predefinido Φ_i. El peso nominal se utiliza para definir el tamaño de la ventana de transmisión de ese ONU (en bytes) como se muestra en la siguiente ecuación:

$$W_i = \frac{\Phi_i}{\sum_{j=1}^{N} \Phi_j} W_{MAX} \quad (1)$$

donde W_{MAX} es el tamaño máximo de la ventana de transmisión que corresponde al tiempo máximo de ciclo. El ciclo es el intervalo de tiempo entre dos trasmisiones sucesivas de un mismo ONU. En ese período los demás ONUs habrán transmitido también una vez en su ranura de tiempo asignada siguiendo el mismo orden (*round robin*). El OLT puede garantizar entonces un tamaño de ventana de transmisión nominal de $\Phi_i * W_{MAX}$, si:

$$\sum_{j=1}^{N} \Phi_j = 1 \quad (2)$$

El algoritmo DDSPON se puede describir como sigue:

1. EL OLT recibe un mensaje Report proveniente del ONU i, que contiene dos valores: $R_i(n)$ (ventana de transmisión solicitada para el ciclo n) y $\Phi_i(n)$ (el peso calculado para el ciclo n, de acuerdo a lo solicitado). El OLT entonces actualiza su vector de pesos local modificando $\Phi_i(n)$ de la siguiente forma:

ONU_1	ONU_2	ONU_3	……	ONU_N
$\Phi_1(n)$	$\Phi_2(n)$	$\Phi_3(n)$	……	$\Phi_N(n)$

otorgando $R_i(n)$ tal como se ha solicitado mediante un mensaje Gate, e incluyendo el vector previo en el mensaje. Como se verá más adelante, en realidad solo se requiere la suma de todos los valores de este vector para hacer los cálculos en la ONU. Esta sería una forma de disminuir la sobrecarga en el mensaje especialmente para el caso de tener varias clases de servicio, para lo cual sí sería necesario un vector. Desde el punto de vista teórico, en el caso de planificar M clases de servicio, los mensajes Gate incluirían una matriz NxM como la que sigue:

$$\Phi(n) = \begin{pmatrix} \Phi_{11}(n) & \cdot & \Phi_{1M}(n) \\ \cdot & \cdot & \cdot \\ \Phi_{N1}(n) & \cdot & \Phi_{NM}(n) \end{pmatrix} \quad (3)$$

Para esta descripción y por simplicidad, hemos considerado M=1. Sin embargo el caso podría extenderse directamente.

2. Cuando el ONU i recibe el mensaje Gate, éste transmite los datos que están esperando en cola hasta la cantidad especificada por la ventana otorgada. A través del Gate, el ONU obtiene un nuevo vector de pesos, actualiza su propio valor de peso al valor nominal Φ_i y a partir de allí calcula la ventana máxima que tal ONU puede tomar en el siguiente ciclo n+1:

$$W_i(n+1) = \frac{\Phi_i}{\sum_{j=1}^{N} \Phi_j(n)} W_{MAX} \quad (4)$$

El ONU también calcula el tamaño de ventana de transmisión de la siguiente forma:

$$R_i(n+1) = MIN\left(W_i(n+1), Q_i\right) \quad (5)$$

donde Q_i es el tamaño de la cola en el ONU i en el momento del cálculo. Finalmente, el ONU envía al OLT (mediante el mensaje Report) dos valores: $R_i(n+1)$, y el peso para del ciclo siguiente calculado como sigue:

$$\Phi_i(n+1) = \frac{R_i(n+1)\sum_{j=1}^{N} \Phi_j(n)}{W_{MAX}} \quad (6)$$

donde la sumatoria de los pesos en (6) es la misma usada en (4).

El proceso se reitera del mismo modo para cada ONU. En el punto inicial de operación, el peso del ONU i se establece con el valor nominal Φ_i, que garantiza el ancho de banda acordado en el SLA al inicio de la conexión.

Se debe notar que la planificación se efectúa de forma instantánea (*on-the-fly*). En este caso, no se necesita esperar a recibir todos los mensajes Report en el OLT para proceder a ejecutar el algoritmo DBA, como es el caso de muchos esquemas centralizados. Para este artículo hemos escogido un algoritmo centralizado eficiente (decisión instantánea) IPACT para compararlo al esquema propuesto, y que además es el más conocido y referencia en trabajos de esta temática.

También cabe mencionar que el ONU asignará el tamaño preciso en bytes que pueden entrar en la ventana máxima de transmisión al computar la ecuación (5). En los esquemas centralizados, es el OLT quien trunca el valor máximo a asignar, pero éste no conoce si esto truncará también un paquete Ethernet, resultando en una subutilización del canal. Es por ello que no se necesita informar acerca de posibles umbrales relativos a los posibles valores donde el OLT puede proceder a truncar el valor del tamaño de la ventana de transmisión sin incurrir en desaprovechamiento del medio.

Aquí el ONU es quien planifica dinámicamente el tamaño de su ranura de tiempo, escogiendo un valor que no afecte la transmisión de alguna trama Ethernet, las cuales no pueden fragmentarse en la red EPON.

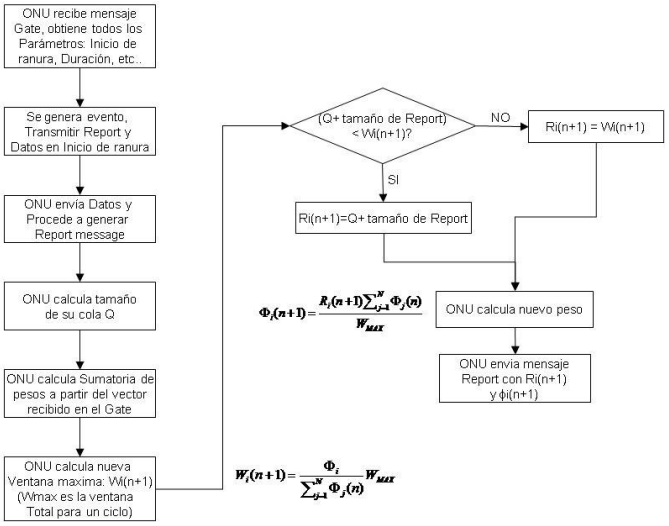


Fig. 1. Diagrama de Flujo para el ONU

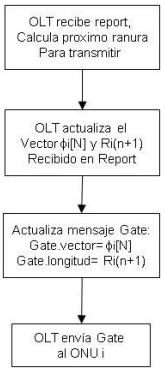


Fig. 2. Diagrama de Flujo para el OLT

En las figuras 1 y 2 se muestran los diagramas de flujo del algoritmo a ejecutarse en el ONU y en el OLT respectivamente. Se puede ver que la parte computacional en el ONU es más extensa que en el OLT, sin embargo, los procesos o cálculos requeridos son simples de implementar en los nodos ONU.

IV. DESARROLLO DE UN SIMULADOR DEL ESTÁNDAR IEEE 802.3AH EPON

Para evaluar el comportamiento del algoritmo DDSPON, se realizaron numerosas simulaciones basadas en el entorno de simulación *OPNET Modeler* bajo el cual se ha desarrollado un modelo de red óptica pasiva cumpliendo estrictamente el estándar IEEE 802.3ah el cual define las especificaciones de las redes tipo EPON. El modelo de red ha sido desarrollado de tal forma que es posible evaluar las prestaciones de distintos esquemas de asignación de ancho de banda fácilmente, así como la influencia en el comportamiento de la red de distintos parámetros característicos como la distancia, el tiempo de ciclo, etc.

OPNET Modeler es una herramienta que permite la simulación de sistemas de comunicaciones con el fin de evaluar prestaciones bajo diferentes condiciones, es el caso de flujos variables de tráfico, pérdida de paquetes, entre otros parámetros.

El diseño de modelos de red en *OPNET* se hace de forma jerárquica por lo que se parte de tres niveles dentro de los cuales se encuentra: el modelo de red en donde se definen las redes a simular a partir de la interconexión de nodos, el modelo de nodos que define la estructura de cada uno de los componentes de las redes basados en la interconexión de módulos y el modelo de procesos que representa la base del sistema de modelado pues permite programar en lenguaje C++ las funciones de cada módulo que definirán el comportamiento de la red.

OPNET Modeler es una herramienta que permite la simulación de sistemas de comunicaciones bajo diferentes condiciones. Además *OPNET Modeler* cuenta con numerosas librerías y modelos predefinidos que corresponden a modelos de redes con protocolos de comunicación estándar que facilitan el diseño de nuevos modelos que requieran de la incorporación de nuevas características.

Nuestro modelo de red contempla a nivel de red tres unidades: usuarios con hasta 32 nodos (ONU), el OLT con capacidad para 32 unidades y un divisor/combinador óptico.

Para determinar los módulos que constituirán la estructura interna del diseño del ONU, se hizo una distinción de los procesos que éste lleva a cabo y que contempla el estándar IEEE 802.3ah. De esta manera se identificaron los módulos que lo integran permitiendo que el modelo fuese de fácil manipulación para implementaciones futuras. Es así que el ONU quedó formado por un total de 10 módulos a nivel de modelo de nodos, una cola (que simplifique el análisis de las simulaciones pero que fácilmente podría considerar más de una cola), dos receptores y un transmisor.

El OLT comprende un total de 13 módulos un receptor y un transmisor. Es importante señalar que el modelo de red diseñado supone la fácil adaptación de nuevos algoritmos de asignación de ancho de banda con la modificación de uno de los módulos implementados. El modelo del dispositivo óptico, entendido como un combinador óptico en el canal de subida y un divisor óptico en el canal de bajada es simple en su diseño esta formado por 2 módulos uno por cada canal de recepción, 33 receptores y 33 transmisores.

El desarrollo del modelo de red consistió en el diseño de cada uno de los componentes de una red de acceso tipo EPON con tecnología de acceso TDM (*Time Division Multiplexing*) y la evaluación del entorno de simulación bajo el esquema de asignación de ancho de banda de ventana fija. Posteriormente se implementó bajo el esquema DDSPON y el esquema IPACT. De esta manera ha sido posible evaluar las prestaciones de la red. Así también ha sido posible analizar el comportamiento y validar la efectividad de DDSPON.

Diferentes escenarios han sido analizados, en los que parámetros como la disposición de los ONUs así como atributos del tráfico han sido modificados. La Fig. 3. muestra el diagrama general de la red en el cual se considera una topología de tipo árbol con 16 ONUs, cada uno de ellos separados del OLT a una distancia de 10km a 20km en general.

Las simulaciones comparan el desempeño del algoritmo DDSPON e IPACT, este último basado en el método de servicio limitado, es decir, el OLT asigna el número de bytes solicitado pero dicha asignación no supera la ventana máxima de transmisión predefinida.

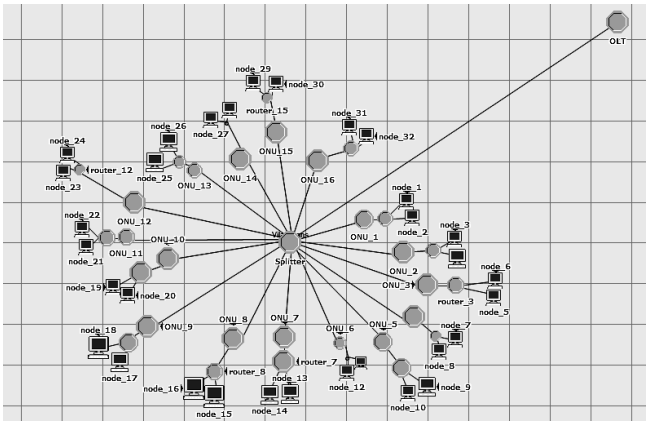


Fig. 3. Modelo de Red.

V. SIMULACIONES Y RESULTADOS

Para llevar a cabo un análisis de las prestaciones más real se utiliza un modelo de tráfico auto-similar ya que con este modelo de tráfico es posible caracterizar los flujos de tráfico de la mayoría de aplicaciones en redes de comunicaciones. Para generar el tráfico auto-similar, *OPNET Modeler* provee un modelo de generación de tráfico denominado *Raw Packet Generator* RPG en el cual los atributos del tráfico auto-similar son definidos es el caso del tamaño del paquete, la tasa media de llegadas y el parámetro de Hurst (H) que fue considerado en este estudio de H=0,7 y H=0,8. El tamaño medio del paquete sigue una distribución uniforme para un tamaño aleatorio del paquete con un límite inferior de 64 bytes y un límite superior de 1518 bytes correspondiente al tamaño máximo de un paquete Ethernet.

Para obtener resultados correspondientes a diferentes cargas de la red, donde el total de la carga ofrecida es de 1 Gbps y es distribuida equitativamente entre todos los ONUs activas, la tasa media de llegadas varía proporcionalmente de acuerdo a la carga de la red que se evalúe en la simulación.

Las simulaciones se han realizado con diferente número de semillas de tal forma que las muestras obtenidas permitan aproximar el valor de la media al valor real de la misma.

Las estadísticas recolectadas son principalmente los valores medios de: tamaño de cola (Q), y retardo del paquete, calculado como la suma de tres componentes: el retardo en el *polling*, es decir el intervalo de tiempo que transcurre desde la llegada de un paquete y la siguiente solicitud de ancho de banda por parte del ONU, el retardo en el *grant* (o mensaje Gate) que se refiere al tiempo desde que se envía una solicitud de ancho de banda hasta que se recibe un mensaje de *grant* que indica el tamaño de la ventana de transmisión concedido, y finalmente el retardo de la cola que indica el tiempo que permanece el paquete en la cola hasta su transmisión. La Fig. 4. representa los tres componentes del retardo de los paquetes.

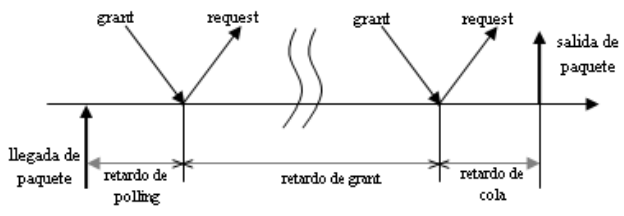


Fig. 4. Componentes del retardo de paquetes

Se consideraron cinco escenarios de simulación donde se han variado parámetros como la distancia entre ONUs y OLT así como el tráfico. Para todos los escenarios se considera la EPON de 1 Gbps con 16 ONUs, cada ONU con un enlace hacia los usuarios de 100Mbps y una cola de tamaño infinito. El intervalo de guarda es de 0,008ms y el tamaño máximo del ciclo se define de 1ms. La distancia entre el OLT y los ONUs se dispone tomando en cuenta la distancia máxima de las EPON, es así que se consideran distancias largas o máximas de 20km, medias de l0km y distancias cortas de 5km aproximadamente. Los escenarios 1 y 2 representan las distancias máximas, el escenario 4 las medias y el escenario 5 las cortas. El escenario 3 sin embargo representa la disposición de ONUs a diferentes distancias entre medias y largas, es decir que la mitad de ONUs se ubican a media distancia y el resto a máxima distancia, de esta manera podremos evaluar el comportamiento de los algoritmos en casos donde los ONUs estén localizados a diferentes distancias entre máximas y medias.

Además se debe tener en cuenta que las simulaciones se realizaron sin considerar el proceso de descubrimiento de los ONUs de forma que únicamente las cuestiones directamente relacionadas con el algoritmo de asignación de ancho de banda han sido analizadas. La duración de las simulaciones fue de 0,3 segundos, tiempo con el cual se obtuvieron resultados válidos que permiten evaluar el comportamiento de los algoritmos.

La Tabla 1 muestra los parámetros considerados en el conjunto de simulaciones.

Parámetros	Esc. 1	Esc. 2	Esc. 3	Esc. 4	Esc. 5
Número de ONUs	16	16	16	16	16
Tasa de enlace de usuario a ONU	100Mbps	100Mbps	100Mbps	100Mbps	100Mbps
Tasa EPON	1Gbps	1Gbps	1Gbps	1Gbps	1Gbps
Número de colas por ONU	1	1	1	1	1
Tamaño de almacenamiento de la cola	Infinito	Infinito	Infinito	Infinito	Infinito
Intervalos de guarda	.008ms	.008ms	.008ms	.008ms	.008ms
Tamaño máximo del ciclo	1ms	1ms	1ms	1ms	1ms
Distancia en KM entre OLT y ONU	18<d<20	18<d<20	10<d<20	10<d<11	4<d<5
Parámetro Hurst	H=0,7	H=0,8	H=0,8	H=0,8	H=0,8

Tabla 1. Parámetros de simulación

Por simplicidad de análisis, las simulaciones solo se han considerado con una cola por ONU, sin embargo se puede extender hasta 8 colas de acuerdo al IEEE 802.3ah.

Las estadísticas fueron evaluadas para las diferentes cargas de la red por lo que los valores medios obtenidos en cada escenario fueron visualizados con el fin de estudiarlos en función de la carga ofrecida.

Los resultados de comparar los primeros dos escenarios en donde el parámetro de Hurst es diferente, se muestran en la Fig. 5. Se observa que el tamaño medio de la cola en el escenario 1 y escenario 2 es muy similar (para un mismo DBA) a diferentes cargas de la red, por lo que estos valores del parámetro de tráfico Hurst no afectan significativamente al comportamiento de la EPON.

Ambos algoritmos experimentan un comportamiento similar para cargas bajas de la red; sin embargo, cuando la carga de la red es mayor a 0,7 el algoritmo IPACT comienza a experimentar un incremento en el tamaño de la cola mayor a la de DDSPON. Cuando la carga de la red alcanza su máxima capacidad, IPACT presenta un valor medio (que es el máximo de ambos escenarios) de aproximadamente 428

Kbytes, mientras que DDSPON alcanza tan solo un valor máximo de 148 Kbytes. Estos resultados tienen implicaciones en el tamaño del almacenamiento de la cola puesto que en el caso de IPACT este requerirá de mucha mayor capacidad respecto a DDSPON para evitar la pérdida de paquetes. Los parámetros iniciales de las simulaciones consideran un tamaño de almacenamiento infinito que nos permitirá conocer el tamaño de cola más idóneo en la práctica.

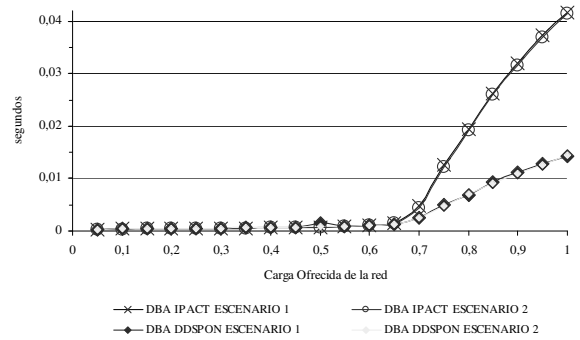


Fig. 5. Retardo medio por paquete

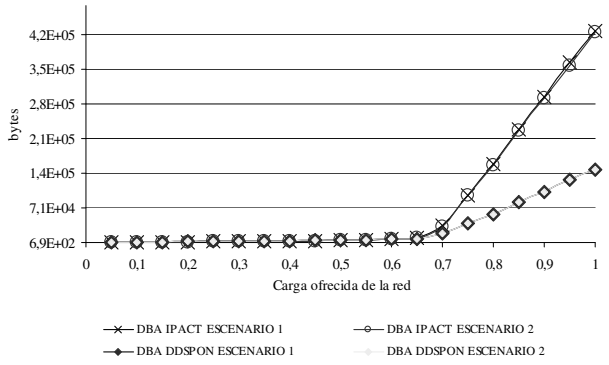


Fig. 6. Tamaño medio de la cola

La Fig. 6. representa el retardo medio por paquete y muestra el incremento del retardo cuando la carga de la red es superior a 0,7. Cuando la cola incrementa de forma notoria su tamaño, también se refleja el incremento en el retardo del paquete. DDSPON muestra menores retardos para altas cargas de la red comparado con IPACT. Como se puede observar en este último el retardo alcanza, para la máxima capacidad de la red, un valor de 0,041 segundos, mientras que por el contrario el retardo del DDSPON es menor de 0,014 segundos.

El nivel de confianza de los resultados de las simulaciones están basados en una Distribución T puesto que el número de muestras es inferior a 30, por lo que con un 95% de confianza el tamaño promedio de la cola se localiza conforme lo que se muestra en la Tabla 2 para IPACT y Tabla 3 para DDSPON.

Carga Ofrecida	INTERVALOS DE CONFIANZA	
	IPACT ESCENARIO 1	IPACT ESCENARIO 2
0,05	859 ± 53	942 ± 52
0,2	1818 ± 16	1809 ± 14
0,4	3302 ± 16	3292 ± 13
0,6	7058 ± 35	7078 ± 38
0,8	158655 ± 4223	157538 ± 5268
1	427966 ± 4807	425367 ± 4952

Tabla 2. Tamaño medio de la cola (bytes). IPACT

Carga Ofrecida	INTERVALOS DE CONFIANZA	
	DDSPON ESCENARIO 1	DDSPON ESCENARIO 2
0,05	726 ± 99	694 ± 59
0,2	1849 ± 17	1890 ± 16
0,4	3512 ± 29	3497 ± 31
0,6	6852 ± 70	6766 ± 90
0,8	57266 ± 3080	59307 ± 3134
1	146609 ± 3804	148205 ± 4004

Tabla 3. Tamaño medio de la cola (bytes). DDSPON

Respecto al retardo medio por paquete se tienen los siguientes valores que se muestran a continuación para IPACT y DDSPON respectivamente.

Carga Ofrecida	INTERVALOS DE CONFIANZA	
	IPACT ESCENARIO 1	IPACT ESCENARIO 2
0,05	0,000283335 ± 2,26235E-06	0,000285718 ± 2,47015E-06
0,2	0,000373628 ± 1,20602E-06	0,000372799 ± 8,28567E-07
0,4	0,00056845 ± 1,34944E-06	0,000567276 ± 1,18293E-06
0,6	0,001048621 ± 2,90793E-06	0,001050316 ± 3,4482E-06
0,8	0,019284324 ± 0,000495513	0,019094532 ± 0,000571013
1	0,041620252 ± 0,000400078	0,041337469 ± 0,000377795

Tabla 4. Retardo medio por paquete (segundos). IPACT

Carga Ofrecida	INTERVALOS DE CONFIANZA	
	DDSPON ESCENARIO 1	DDSPON ESCENARIO 2
0,05	0,000289528 ± 5,68558E-06	0,000290251 ± 4,96606E-06
0,2	0,000406279 ± 5,49087E-06	0,000411083 ± 4,27911E-06
0,4	0,000622077 ± 3,07429E-06	0,000622161 ± 4,45099E-06
0,6	0,001015063 ± 9,02648E-06	0,001000875 ± 9,62725E-06
0,8	0,006864928 ± 0,000356366	0,007091637 ± 0,000353267
1	0,014178433 ± 0,00029868	0,014310852 ± 0,000366334

Tabla 5. Retardo medio por paquete (segundos). DDSPON

Los siguientes resultados consideran el mismo parámetro de Hurst, pero variando ahora la distancia entre ONU y OLT, con esto será posible analizar la sensibilidad de los algoritmos respecto a la distancia. Para DDSPON, la Fig. 7 muestra la relación entre el tamaño medio de la cola y la carga ofrecida de la red en cuatro escenarios. Se puede observar que aproximadamente en todos los casos, el comportamiento se mantiene (con pequeñas variaciones), comprobando la estabilidad del esquema propuesto.

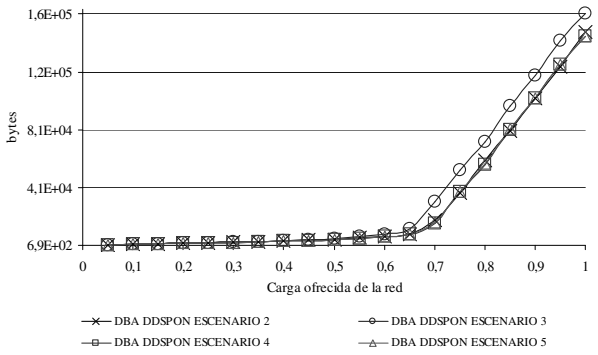


Fig. 7. Tamaño medio de la cola

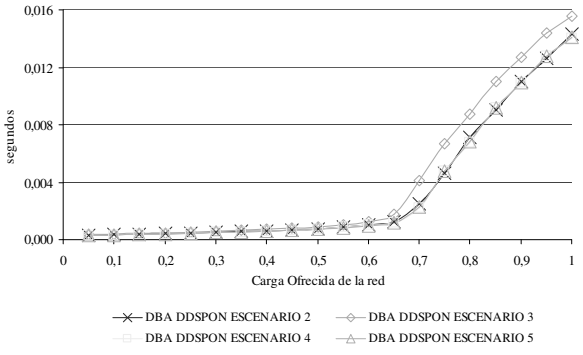


Fig. 8. Retardo medio por paquete

Como se puede observar en el escenario 3, donde los ONUs se encuentran desplegados a distancias medias y largas; es decir de entre 10km y 20km respecto del OLT, DDSPON alcanza los valores más elevados.

De igual forma se obtiene el retardo medio por paquete en relación a la carga de la red que se muestra en la Fig. 8. El máximo valor en cuanto al retardo para DDSPON se observa en el escenario 3 donde, para la máxima carga de la red, se tiene un retardo de 0,015 segundos.

Los valores obtenidos para ambos algoritmos se muestran en las Tablas 6-9. Se puede también apreciar los intervalos de confianza con un nivel del 95%. Los valores obtenidos en los distintos escenarios para DDSPON reflejan una menor variación en los resultados con respecto a los valores obtenidos del algoritmo IPACT, por lo que podemos apreciar que DDSPON mantiene su eficiencia en especial para las altas cargas de la red independientemente de la distancia en la que se encuentran los ONUs.

En las gráficas de la Fig. 9 se puede visualizar una comparación de los tamaños medios de las colas tanto para DDSPON como para IPACT utilizando valores críticos de carga de la red (tales como 0,8 y 1). El peor caso se presenta para DDSPON con el escenario 3, mientras que IPACT lo hace en el escenario 2 donde los ONUs están dispuestos a la distancia máxima. En todos los escenarios DDSPON muestra significativas mejoras en las prestaciones comparado con IPACT.

Carga Ofrecida	INTERVALOS DE CONFIANZA		
	IPACT ESCENARIO 3	IPACT ESCENARIO 4	IPACT ESCENARIO 5
0,05	898 ± 115	814 ± 114	827 ± 111
0,2	1680 ± 218	1683 ± 214	1685 ± 233
0,4	2946 ± 387	2954 ± 398	2948 ± 390
0,6	5513 ± 752	5629 ± 758	5614 ± 748
0,8	103050 ± 15140	102879 ± 15116	102992 ± 15130
1	354008 ± 50343	353821 ± 50330	353418 ± 50285

Tabla 6. Tamaño medio de la cola (bytes). IPACT

Carga Ofrecida	INTERVALOS DE CONFIANZA		
	DDSPON ESCENARIO 3	DDSPON ESCENARIO 4	DDSPON ESCENARIO 5
0,05	722 ± 76	699 ± 56	639 ± 74
0,2	2032 ± 78	1840 ± 22	1833 ± 35
0,4	3914 ± 96	3386 ± 28	3396 ± 37
0,6	7955 ± 171	6816 ± 74	6647 ± 67
0,8	72506 ± 6929	56616 ± 3369	56890 ± 3437
1	161325 ± 8730	144941 ± 4307	145151 ± 4387

Tabla 7. Tamaño medio de la cola (bytes). DDSPON

Carga ofrecida	INTERVALOS DE CONFIANZA		
	IPACT ESCENARIO 3	IPACT ESCENARIO 4	IPACT ESCENARIO 5
0,05	0,00028189 ± 2,7101E-06	0,000265692 ± 4,29565E-06	0,000266151 ± 4,22846E-06
0,2	0,000360284 ± 3,13444E-06	0,000359492 ± 1,69056E-06	0,000361588 ± 4,69176E-06
0,4	0,000526877 ± 2,67898E-06	0,000529089 ± 2,4047E-06	0,000527636 ± 2,87499E-06
0,6	0,000841807 ± 4,56266E-06	0,000860212 ± 3,54181E-06	0,000856417 ± 5,00712E-06
0,8	0,012538025 ± 0,001644225	0,012519411 ± 0,001638312	0,012530129 ± 0,001646676
1	0,034446784 ± 0,004711822	0,034430723 ± 0,004699723	0,034395309 ± 0,004691306

Tabla 8. Retardo medio por paquete (segundos). IPACT

Carga ofrecida	INTERVALOS DE CONFIANZA		
	DDSPON ESCENARIO 3	DDSPON ESCENARIO 4	DDSPON ESCENARIO 5
0,05	0,000333972 ± 2,46063E-05	0,000271337 ± 3,13787E-06	0,000267456 ± 4,27156E-06
0,2	0,000474433 ± 3,04895E-05	0,000386798 ± 1,58567E-06	0,000385781 ± 1,87386E-06
0,4	0,000725188 ± 2,76556E-05	0,00058892 ± 2,5129E-06	0,000590373 ± 3,60742E-06
0,6	0,001214209 ± 3,14266E-05	0,001006371 ± 8,51185E-06	0,000979813 ± 8,00483E-06
0,8	0,008701238 ± 0,000788819	0,006787005 ± 0,000372766	0,006816784 ± 0,000384658
1	0,015592914 ± 0,000797015	0,014029764 ± 0,000396848	0,014059423 ± 0,000403095

Tabla 9. Retardo medio por paquete (segundos). DDSPON

De la Fig. 9 se pueden observar tres aspectos interesantes. Primero se puede verificar con estas gráficas que el esquema DDSPON mantiene niveles de tamaño medio de la cola mucho menores que en el caso del esquema IPACT. En segundo lugar, vemos que DDSPON se mantiene más estable frente a las variaciones de las distancias, mientras que IPACT, especialmente para mayores distancias, presenta una gran variación respecto a otros escenarios. Finalmente es interesante notar que DDSPON presenta un incremento en el tamaño medio de las colas cuando las distancias entre ONUs y OLT son más dispares (entre 10 y 20 Km) como es el caso del escenario 3. Esta variación es de 10 a 21% mayor (para cargas de 0,8 y 1 respectivamente) comparado con los demás escenarios donde las distancias son más homogéneas. Sin embargo, al compararse con el esquema centralizado este aspecto queda minimizado, y las prestaciones de DDSPON siguen siendo muy superiores. Estas pequeñas variaciones en DDSPON se deben a las diferencias de los tiempos de retorno (RTT) que pueden afectar en la total adquisición del estado de la red para algunos ONUs, lo cual conlleva a un incremento en el tiempo de espera en cola, y por ende de su tamaño a causa de la imprecisión en la información sobre el estado de la red.

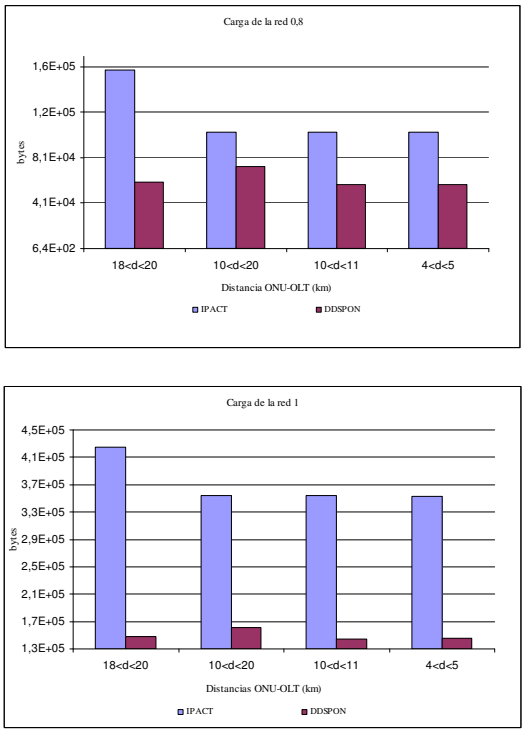


Fig. 9. Tamaño medio de la cola en DDSPON e IPACT para valores normalizados de carga de la red 0,8 y 1

Similares comentarios se pueden deducir de los resultados en cuanto retardo medio por paquete. En la Fig. 10 se muestra un diagrama de barras que representa la diferencia porcentual que DDSPON ofrece como mejora en términos de retardo medio por paquete respecto a IPACT. La diferencia de retardo medio por paquete en términos porcentajes en contraste con el esquema IPACT, varía de 30,6% a 65,4%. Esto representa una mejora considerable de DDSPON si se compara con el esquema centralizado.

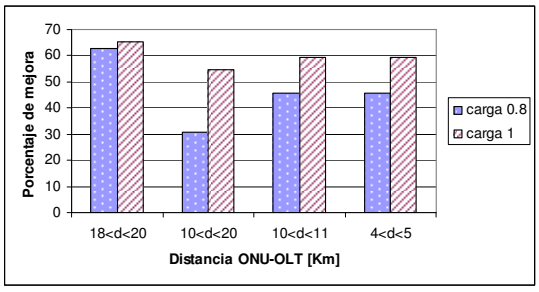


Fig. 10. Retardo medio por paquete: Porcentaje de mejora que ofrece DDSPON sobre el esquema IPACT

Los resultados de las simulaciones permiten pues confirmar el buen comportamiento del algoritmo DDSPON especialmente para altas cargas de la red comparado con el esquema centralizado de referencia en todos los escenarios.

VI. CONCLUSIONES

Se ha presentado un esquema para la asignación dinámica de ancho de banda, el cual como característica diferencial respecto a otras propuestas es del tipo distribuido, es decir, se ejecuta en los dispositivos de usuarios y acorde a la carga real del mismo y del resto de la red. En este trabajo se presenta un estudio comparativo de los parámetros de interés utilizando modelos de simulación que los autores han desarrollado sobre la herramienta OPNET. El modelo implementado cumple con los requerimientos y máquinas de estado expresados en el estándar de EPON. Se ha podido validar con este trabajo otros estudios preliminares del algoritmo DDSPON y se ha podido verificar su eficiencia respecto al esquema centralizado de referencia. Los resultados de las simulaciones también demuestran una mayor estabilidad de la red EPON utilizando el algoritmo DDSPON cuando se evalúa el sistema ante diferentes distancias entre el dispositivo de usuario y el dispositivo de red; a diferencia del esquema centralizado el cual reduce su eficiencia de forma significativa cuanto mayor es la distancia.

AGRADECIMIENTOS

Este proyecto ha sido desarrollado dentro del marco del los proyectos TSI2005-06092, TSI2006-12507-C03-03 y EURO-NF.

REFERENCIAS

[1] G. Kramer y G. Pesavento, “Ethernet passive optical network (EPON): building a next generation optical access network”, IEEE Communications Magazine, Febrero 2002
[2] IEEE 802.3ah : http://www.ieee802.org/3/efm/public.
[3] G. Kramer, B. Mukherjee y G. Pesavento, “IPACT: a dynamic protocol for an Ethernet PON (EPON)”, IEEE Communications Magazine, Febrero 2002
[4] OPNET Modeler : http://www.opnet.com/
[5] G. Kramer, B. Mukherjee y G. Pesavento, “Ethernet PON (ePON): Design and Analysis of an Optical Access Network”, Photonic Network Communications, Julio 2001
[6] M. Ma, Y. Zhu y T. Cheng, “A Bandwidth Guaranteed Polling MAC Protocol for Ethernet Passive Optical Networks”, INFOCOM 2003, IEEE, Marzo 2003
[7] Y. Hsueh, F. An, K. Kim, y L. Kazovsky, “A New Media Access Control Protocol with Quality of Service and Fairness Guarantee in Ethernet-based Passive Optical Networks”, 2nd Symposium on Photonics, Networking, and Computing, Septiembre 2003
[8] S. Choi, “Cyclic Polling-Based Dynamic Bandwidth Allocation for Differentiated Classes of Service in Ethernet PassiveOptical Networks”, Photonics Network Communications, Enero 2004
[9] C. Assi, Y. Ye, S. Dixit y M. Ali, “Dynamic Bansdwidth Allocation for Quality-of-Service Over Ethernet PONs”, IEEE Journal on Selected Areas in Communications, Noviembre 2003
[10] C. Chae, E. Wong y R. Tucker, “Optical CSMA/CD media access scheme for Ethernet over passive optical network”, IEEE Photonics Tchnology Letters, Mayo 2002.
[11] Y. Tse, L. Chen y C. Chan, “A Distributed Collision Avoidance Protocol using Pilot Tone-based Carrier Sense Mechanism for Ethernet Passive Optical Networks”, Optical Fiber Communications Conference, 2003, OFC 2003, Marzo 2003
[12] C. Foh, L. Andrew, M. Zukerman, and E. Wong, “FULL-RCMA: a high utilization EPON”, Optical Fiber Communications Conference, 2003, OFC 2003, Marzo 2003
[13] S. Sherif, A. Hadjiantonis, G. Ellinas, C. Assi y M. Ali, “A Novel Decentralized Ethernet-based PON Access Architecture for provisioning Differentiated QoS”, Journal of Lightwave Technology, vol. 22, pp. 2483-2497, 2004
[14] M. De Andrade, L. Gutierrez y S. Sallent, “DDSPON: A Distributed Dynamic Scheduling for EPON”, IEEE International Conference on Signal Processing and Communication (ICSPC 2007), Noviembre 2007

Purpose of a modified Pathchirp method for available bandwidth computing in an end to end path

Yury Andrea Jiménez Agudelo, Sebastia Sallent Ribes, Cristina Cervelló Pastor
Department of Telematic Engeneering,
Polytechnic University of Catalonia
Canal Olimpic 15, Castelldefels 08860, Spain
yuryandrea@gmail.com, sallent@entel.upc.edu.

***Abstract*- This paper presents a comparative study about two active probing methods -SLoP and Pathchirp- for estimating the available bandwidth on an end to end path network, these methods are based on the concept of self induced congestion. From this study we purpose a modified Pathchirp method, where the main result obtained with the purpose method is the lower response time and load in the network than those presented in SLoP and Pathchirp methods. This method improves significantly the accuracy of the measurements without higher the load in the network compared to SLoP and Pathchirp.**
The purpose probing scheme is ideal due to it provides an accurate estimation of a path available bandwidth in a time as short as possible, while imposing a load as low as possible on the network.
t
***keywords*- probing, SLoP, Pathchirp, modified pathchirp, available bandwidth, self-induced congestion, end to end.**

I. INTRODUCTION

The estimation of available bandwidth is very important in the packet networks due to it establishes the relationship of the information amount that a network path can deliver per unit of time (bps). In relation to quality of service QoS aspects, relevant issue to this paper, the available bandwidth is defined as the minimum needed bandwidth to provide the QoS requirements such as delay, jitter and packet losses for a data flow.

Knowing the available bandwidth on an end-to-end path several network applications or traffic control mechanism can be improved, such as rate-based streaming applications [1], admission control [2], congestion control [3], as well as service level agreement verification (SLA) among other and also augments applications as diverse as grid computing and overlay networking. Finally, real time information about to available bandwidth aids network operations managers in different operational tasks such configuring traffic routers.

At present, it is very difficult to obtain accurate estimations of the available bandwidth directly from internet routers because of the internet decentralized nature so that the measures are insufficient accurate, technical called active method. Thus, another alternative way for obtaining required information from the network edge is by using passive available bandwidth computing methods such as PathChirp and SLoP among others. Edge-based measurements are the best option for inferring the congestion state of a path network at interval time [4].

In this paper, we present a modified of the edge-based active probing tool Pathchirp used to locate the less available bandwidth link commonly called “tight link” in an end to end path. This modified provides an accurate estimation of the path’s available bandwidth in a short time and without increase the load; in this case the time is reduced significantly.

II. RELATED WORK

Several of today’s available bandwidth estimation fall into two classes. The first class is scheme is based on statistical cross traffic models, such as Delphi [5] and other methods which are proposed in [6]. These schemes are characterizes by provide accurate estimates of cross traffic but this scheme only estimate bandwidth for single hop so that this scheme is not robust in path end to end, which aren’t represent the real scenario.

The second scheme is based on the concept of self-induction congestion, these schemes can available bandwidth estimates for end to end path, examples include Trains of Packet Pairs (TOPP) [7], Initial Gap Increasing (IGI) [8], Pathload or SLoP [9], Network Test [10], and PathChirp [4].
The self-induced congestion principle provides an effective technique for estimating the available bandwidth in a path. This principle relies on the fact that router buffer incoming packets are queued before transmitting them via link. In other words, if the incoming packets bit rate exceeds the outgoing transmission rate of the link, the packets fill up the corresponding queue leading to queuing delays, as shown in figure 1. Such schemes are equally suited to single and multiple hop paths, since they rely only on whether the probe packets make it across the path with an unusual delay or not.

These tools differ from each other in both the type of probe schemes and the algorithms they use. There schemes are divided in two groups, first scheme uses as probe packets a packet train; second scheme uses packet train of just two packets is a packet pair.

TOPP and IGI scheme uses packet pairs with different spaced between them or interspacing them. The probing bit

rate at which packet interspacing at the receiver host begins to exceed that at the sending host gives the available bandwidth.

SLoP method uses packet trains of equally spaced packets, employs long constant bit-rate (CBR) packet trains and adaptively varies the rates of successive packet trains in an effort to converge to the available bandwidth rate, which allows be a very accurate method.

PathChirp's algorithm uses packet trains with different spaced between packets, the spacing between successive packets decreases exponentially according to a spread factor γ.
By using a few probe packets, a chirp can thus sweep through a wide range of probing rates, enabling a quick estimate of the available bandwidth.

In this study were selected the SLoP and PathChirp methods because these methods has advantages on other as such accuracy and efficiency respectively.

III. PATHCHIP AND SLOP METHOD

According to the self-induced congestion principle, if probe packets are sent through a network path at a bit rate R faster than the available bandwidth B, R > B, then the path's queues will congest, increasing delays or loss packets. On the other hand, the path's queues won't congest if the bit rate is less than available bandwidth, R < B. Pathchirp and SLoP estimate the available bandwidth varying the sent probing bit rate to identify the minimum rate at which increasing packet queuing delays q_k are detected.

The self-induced congestion principle is used to establish the relation between the transmission rate and available bandwidth as follow:

$$\begin{array}{lll} R \geq B & if & q_k \geq q_{k+1} \\ R < B & if & q_k \leq q_{k+1} \\ R \infty B & if & \Delta q_k \equiv 0 \end{array} \quad (1)$$

Where q_{k+1} is the following queuing delays after of q_k.

In the Figure 1 are shown the cases described in (1), where the congestion node which has an increasing packet queuing delays and the other nodes present queuing delay closely zero, this nodes no present congestion. The former are congestion free.

A. Self-Loading Periodic streams – SloP

SLoP employs long constant bit-rate (CBR) packet trains and adaptively varies the rates of successive packet trains in an effort to converge to the available bandwidth rate. The algorithm is shown in 2. Because of its adaptive search, SLoP can have long convergence times (on the order of 100s of RTTs) [11] and use a great amount of probe traffic.

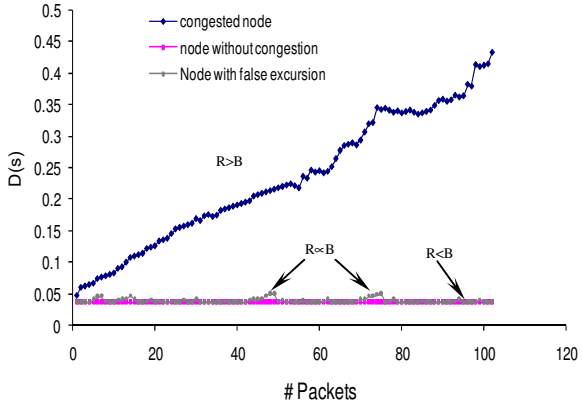


Fig. 1. Features of the self-induced congestion principle.

$$\begin{array}{l} If\ R(n) > B, \quad R^{\max} = R(n); \\ If\ R(n) \leq B, \quad R^{\min} = R(n); \\ R(n+1) = (R^{\max} + R^{\min})/2; \end{array} \quad (2)$$

This algorithm is based on a sender (SND) which provides a periodic stream with K packets to a receiver (RCV) at a rate R_0. The difference between two successive packets k and k+1 at the RCV, denoted as One-Way Delay q^K, is:

$$\Delta q_k \equiv q_{k+1} - q_k$$

$R^{\min}$ y $R^{\max}$ are the lower and upper bounds for the available bandwidth after stream n. Initially, $R^{\min}$=0 and $R^{\max}$ is a value sufficiently close to B.
The algorithm finishes when $R^{max} - R^{min} \leqq \omega$, where ω is an estimated resolution.
SLoP determines the relation between the sent rate and the available bandwidth by analyzing the packets delay at the receiver. If the periodic packet stream presents delay increases regarding to packets send time then R > B. If packets delay is constant it is satisfied that R < B.

Figure 2 shows the available bandwidth estimation time in function of the accuracy. In this figure it is possible see that the higher accuracy the higher computing time, e.g. to achieve an accuracy of 1% it is required a time of 0.1 seg.

This method can be very accurate but it leads to increase the network load and the available bandwidth estimation time.

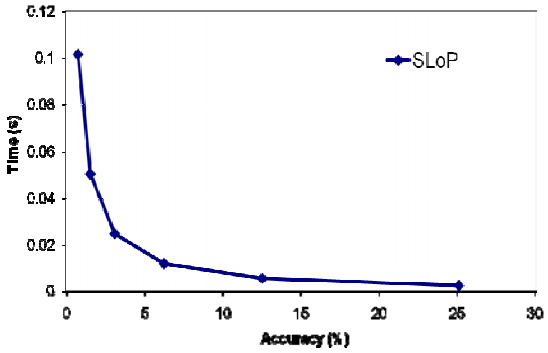


Fig. 2. Estimation time in function of the accuracy for SLoP method.

B. Pathchirp

This method sends a chirp probe train to detect the available bandwidth; the spacing between successive packets decreases exponentially according to a spread factor γ. The first packets interspacing is denoted T and the subsequent packets interspacing are Tγ, $T\gamma^2$, $T\gamma^3$, and so on, as it is shown Figure in 3. A chirp can thus sweep through a wide range of probing rates, enabling a quick estimate of the available bandwidth based on the self-induced congestion. Pathchirp estimates the available bandwidth when the probes packets suffer queuing delays. The delay includes the speed-of-light propagation delay and the packet service time at intermediate queues.

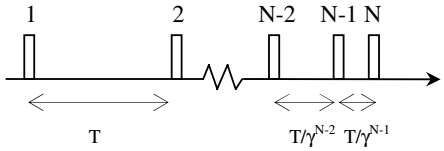


Fig. 3. Features spread factor of a train chirps.

The spread factor γ controls the spectrum of probing rates in a chirp. This factor determines the estimation accuracy of available bandwidth, which generally is between 20% and 40**%** [4]. This factor also determines the number of packets per chirp and hence reduces or increases the number of estimates per time interval, therefore the higher accuracy the higher load and computing time, as it is shown Figure in 5 and 7, respectively.

Figure 4 shows the typical behavior of the queuing delays of a chirp train and the chirp packet that determinates the available bandwidth E_k in the end to end path.

In Figure 4 it can be observed some regions where the delay packets increases, these regions are denominated excursions, those excursions are presented when some chirp packets suffer queuing delays, q > 0.

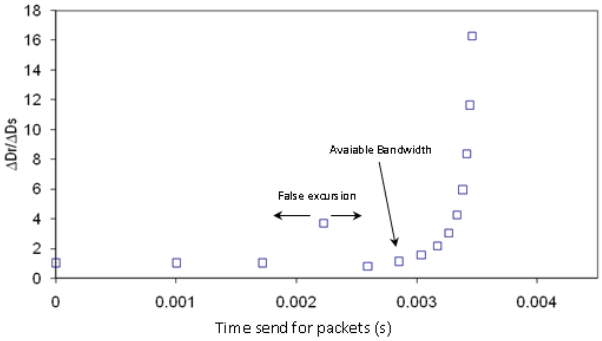


Fig. 4. The profile of a chirp-packets queuing delay.

Also the figure shows the excursions from the zero for several consecutive packets. The first small excursion ends with the queuing delays returning to zero, this is because the chirp rate R is less than the tight link capacity B seeping on the path which is denominated false excursion. The last excursion usually ends with an increasing progressive queuing delay because R > B, which causes the chirp packets to fill up intermediate queues.

Each chirp packet corresponds to an estimation of the available bandwidth; the packet is detected at the receiver by averaging the obtained estimates in the time interval during the last chirp train. After an excursion is detected it is necessary identify if it maintain an increasing tendency or if it is a false excursion.

To find the chirp packet which defines the available bandwidth, all delay packets are evaluated taking into account the next three possibilities [4]:

Case A: If k belongs to an excursion that terminates and $q_k \leq q_{k+1}$.

$$E_k = R_k$$

Case B: If k belongs to an excursion that does not terminate, then set

$$E_k = R_l \quad \forall\ k > l$$

Case C: For all k not belonging to the above cases, this includes all those k not belonging to excursions as well as those with decreasing queuing delay belonging to excursion.

$$E_k = R_l$$

Where, *l* is the excursion start packet.

The final solution D is calculated using a weighted average of the E_k per-packet, applying the next equation:

$$D = \frac{\sum_{k=1}^{N-1} E_k \Delta_k}{\sum_{k=1}^{N-1} \Delta_k}$$

One iteration is enough to compute the available bandwidth in a network path; nonetheless, the estimation could be inaccurate. Using a lower spread factor in the packets train the accuracy could be improved.

A better accuracy (10% or less) is obtained increasing the chirp packets number producing an effect of increases the computing resolution; in other words the number of transmission rate samples is higher. However, a bigger network load is injected and therefore the solution takes more time, as it is shown in Figure 5.

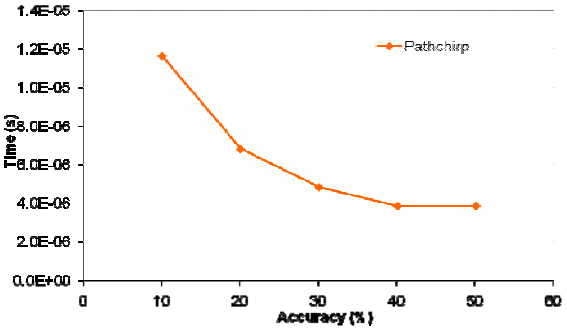


Fig. 5. Features of the Pathchirp method.

Figures 2 and 5 show that SLoP and Pathchirp methods have advantageous features in accuracy and load and time, respectively.

Possible changes in the PathChirp method that permit improve its accuracy which can be similar to that obtained by SLoP method but without introducing high quantities of load are described in the next section.

III. Purpose of a Modified for Computing Pathchirp Method.

To obtaining a better load/accuracy relation, it can be used a method that consists in sending two subsequent chirps with different spread factors, the values of the both factors are defined according to the needed accuracy. The first chirp defines the minimum and maximum limits of an interval that contains the available bandwidth value; the computing accuracy obtained by this first chirp is usually between 10% and 50% (γ= 1.1 to 1.5). In addition the available bandwidth computed by this chirp is used to define the search range limits lower and upper for the second chirp. The second chirp has a lower spread factor typically between 1.01 and 1.1.This last chirp estimates the available bandwidth with a higher accuracy.

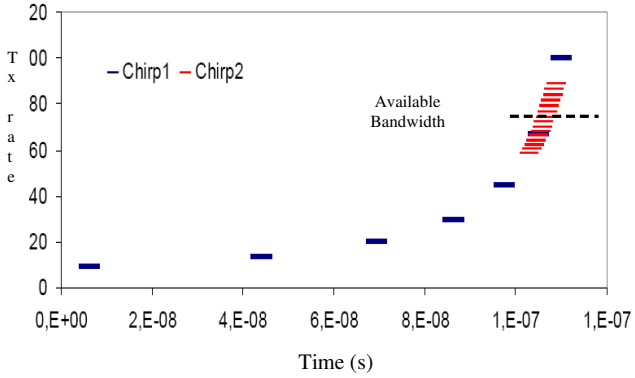


Fig. 6. Features modified Pathchirp method.

In this section will be describe the procedure realized for obtain the dates shown in the tables 1, 2 and 3. The goal is to compare the efficiency of modified method in terms of the number of bytes used to obtain available bandwidth estimates and accuracy with SLoP and PathChirp methods.

For obtained times of transmission end to end for chirps train was used the simulator NS2 version 2.33. The topology network simulated was taken from [12], it network have a series of store-and-forward nodes each with its own constant service rate, equipped FIFO queues of 10Mbps and a communication canal includes the speed-of-light propagation delay and packet service times at intermediate queues.

Were infers available bandwidth sending UDP chirp packet probes from sender to receiver on some paths. These chirp trains found in their path congestion which was simulated with sources of cross-traffic rates. This is an accurate model for today's internet.

Chirp trains travel one-way from sender to receiver, where the receiver performs the available bandwidth estimation on the end to end path, this in order to avoid the problem of echo probe traffic interfering with the send of chirp probes receiver-sender.

The data obtained in the tables were calculated from testing the algorithms to compare the tools. To measure the efficiency of the methods, in each experiment was compute the average number of bytes over 15 runs that each tool takes to provide estimates accurate to a link.

To obtain the bytes used by SLoP, was set its bandwidth resolution parameter to available bandwidth estimation between 1 to 100 Mbps and take the average number of bytes used to make 15 estimates.

To count the bytes used by PathChirp, was calculated the number of probes send in a time interval and then was calculated the number of probes send for reach a specific accuracy for an available bandwidth estimate between 1 to 100 Mbps.

To obtain a given accuracy value many combinations of factors γ_1 and γ_2 which present different load and time values can be used as shown in Tables 1 and 2. Data reported in

Tables 1, 2 and 3 were computed by evaluating of Pathchirp and SLoP algorithms. These algorithms were evaluated on similar conditions for available bandwidth searching in an interval between 1 Mbps and 100 Mbps. The main parameters to evaluate are the total injected load during the available bandwidth searching and the computing time.
The computing time in both methods was found neglecting the receiver response and transmission time.

In the Table 3 the best spread factor combinations that introduce the lowest load in the path are shown. Initially the best combinations were selected depending on the injected load, however, some γ combinations introduce similar loads, therefore a second choose criterion based on the computing time was used. In Tables 1, 2 and 3 the best combinations are remarked.

γ_1 ⋱ γ_2	1.005	1.01	1.03	1.05	1.1
1,1	58400	49600	44000	43200	
1,2	**58400**	**40800**	28800	26400	24800
1,3	73600	44800	**26400**	22400	19200
1,4	95200	54400	27200	**21600**	**17600**
1,5	122400	67200	30400	23200	17600

Table 1. Network injected load (bytes)

γ_1 ⋱ γ_2	1.005	1.01	1.03	1.05	1.1
1,1	11.88	11.76	11.69	11.68	
1,2	**7,33**	**7,09**	6,92	6,89	6,86
1,3	5,72	5,28	**5,01**	4,95	4,90
1,4	5,22	4,56	4,11	**4,02**	**3,96**
1,5	5,89	4,90	4,24	4,11	4,01

Table 2.Computing time (μs)

γ_1/γ_2	Precisión (%)	Carga (bytes)	tiempo (μs)
1.2/1.005	0,5	58400	7,33
1.2/1.01	1	40800	7,09
1.3/1.03	3	26400	5,01
1.4/1.05	5	21600	4,02
1.4/1.1	10	17600	3,96

Table 3. γ combinations features in the modified Pathchirp method

Next two comparatives graphics of the SLoP, PathChirp and modified PatChirp methods performance in relation to load and time are shown. In Figure 7 it can be seen the injected load for an accuracy interval from 0.5 to 50% for each method. In the case of SLoP method a 0.5% accuracy requires a network load close to 0.28MB and around 0.1MB for 25% accuracy. For the case of Pathchirp the highest accuracy reached is close to 10% that generates around 0.05MB network load, whereas 50% accuracy generates a load approximately of 0.01MB. Modified Pathchirp method generates a 0.06MB load with a 50% accuracy and 0.02MB load with an accuracy of 10%.

In the Figure 8 the main features of the computing time versus accuracy for both PathChirp and modified PathChirp methods can be shown, in this graphic it is not shown the SloP method because its computing time is very higher as was illustrate in the Figure 2.

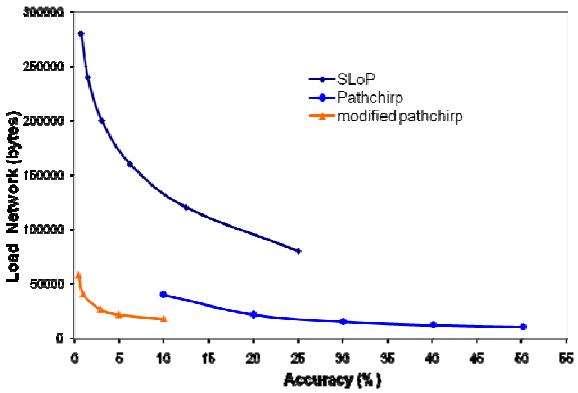


Fig. 7. Features of the Pathchirp, SLoP and modified methods

In the Table 3 is shown that a higher accuracy the higher load similar to the case of SLoP and Pathchirp methods. However the combination of both chirps generates a lower load than Pathchirp.

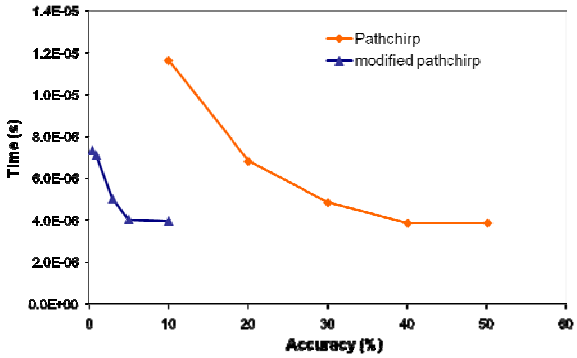


Fig. 8. Features of the self-induced congestion principle.

C. Algorithm of the modified Pathchirp method.

In [4] is described the algorithm for the Pathchirp method which uses a queuing delay vector of the chirp train as the input data and return the available bandwidth in an end to end path.
For the modified Pathchirp method implementation we have developed the following algorithm:
The algorithm starts with the definition of the required accuracy and the queuing chirp packets delay vector. The next parameters are considered constants: probe packets size P, the spread factors γ_1 and γ_2, the decrease factor F, the busy period threshold L and the search bandwidth interval.

The spread factors are calculated from the accuracy defined by the user. The first range in which the available bandwidth is searched corresponds to that defined by the user and the second one is computed from the available bandwidth found by the first chirp execution using the next equations:

$$R_1 = \frac{D}{\gamma_1} \quad ; \quad R_2 = D\gamma_1$$

Where R_1 and R_2 are the limits of the bandwidth interval for the second chirp, this criterion is defined in order to establish a range as high as the packet interspacing of the first chirp.

The estimation function D is similar to that is described in [4] this function gives an estimate of available bandwidth. Table 4 shows the proposed algorithm.

```
Procedure compute_Df(q1, q2 , Acc, Bwend, Bwini) {

/* q1 and q2 denotes the vectors of the chirp train
queuing delays of the two chirps*/

If (Acc<10){
        If (Acc=2) {
                G1 = 1.2 ; G2=1.01
        }
        Else if (Acc=3) {
                G1 = 1.2 ; G2=1.01
        }
        Else if (Acc=5){
                G1 = 1.2 ; G2=1.01
        }
        N1 = (log(Bwend)- log(Bwini))/log (G1);
        D = estimate_D(G1, N);
        R1 = D/ G2 ;
        R2 = D/ G1;
        N2 = (log(R2)-log(R1))/log (G2);
        Df = estimate_D(G2, N2)
}
If(Acc>10){
        G = 1 + Acc/100;
        N1 = (log(Bwend)- log(Bwini))/log (G);
        Df=estimate_D(G, N);
}
return Df;
}
```

Where G1 and G2 correspond to γ1 and γ1.

IV. Conclusions

SLoP and Pathchirp differ in their measurement methodology as well as their output quantities. Pathchirp provides a single estimate of available bandwidth per specified time interval. SLoP instead provides minimum and maximum bounds on the available amount of time to make the estimate. In this paper was shown the main features of these methods.

This paper provides a way to adaptively vary range of chirp probing rates so as to reduce the probing load on the network, obtaining a best performance. The main advantage that provide the purpose modified pathchirp is the higher accuracy, this property allow that it method can to be applied in control mechanism and streaming applications to provide fast estimates.

In the future is needed realize probes that will demonstrate the efficiency of the modified presented in this paper.

Acknowledgements

This project has been developed within the support of the projects TSI2005-06092 y TSI2006-12507-C03-03. Euro-NF.

References

[1] P. Chou and Z. Miao, Rate-distortion optimized streaming of packetized media, in Microsoft Research Technical Report MSR-TR-2001-35, February 2001.

[2] L. Breslau, E. Knightly, S. Shenker, and I. Stoica. Endpoint admission control: Architectural issues and performance, In Proc. ACM SIGCOMM, 2000.

[3] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient overlay networks, In Proc. ACM SOSP, 2001.

[4] V.Ribeiro, R.Riedi. Pathchirp: efficient available bandwidth estimation for network paths. Scientific Discovery through Advanced Computing (SciDAC), 2003.

[5] V. Ribeiro, M. Coates, R. Riedi, S. Sarvotham, B. Hendricks, and R. Baraniuk, "Multifractal cross-traffic estimation," Proc. of ITC Specialist Seminar on IP Traffic Measurement, Sept. 2000.

[6] G. He and J. C. Hou, "On exploiting long-range dependency of network traffic in measuring cross-traffic on an end-to-end basis," IEEE INFOCOM, 2003.

[7] B. Melander, M. Björkman, and P. Gunningberg, "A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks," Proc. IEEE Globecom Global Internet Symp., IEEE CS Press, 2000, pp. 415–420.

[8] N. Hu and P. Steenkiste, "Evaluation and Characterization of Available Bandwidth Probing Techniques," IEEE J. Selected Areas in Comm. Special Issue on Internet and WWW Measurement, Mapping, and Modeling, vol. 21, no. 6, 2003, pp. 879–894.

[9] M. Jain and C. Dovrolis, "End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput," IEEE/ACM Trans. Networking, vol. 11, no. 4, 2003, pp. 537–549.

[10] G. Jin and B. Tierney, "Netest: A Tool to Measure the Maximum Burst Size, Available Bandwidth, and Achievable Throughput," Proc. Int'l Conf. Information Technology, Research and Education (ITRE), 2003; http://dsd.lbl.gov/ DIDC/papers/netest-mbs.pdf.

[11] M. Jain and C. Dovrolis, "End-to-End available bandwidth; measurement methodology, dynamics, and relation with TCP throughput," Proc. ACM SIGCOMM, 2002.

[12] P. Rodriguez, W. Biersack. Dynamic Parallel Access to Replicated Content in the Internet. IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 10, NO. 4, AUGUST 2002.

Cardea. Una plataforma OSGi para Servicios Hospitalarios

Saúl Navarro
Grupo Gesfor
Avda Manoteras, 32 28050 Madrid
snavarrob@grupogesfor.com

Silvia Platas
Grupo Gesfor
Avda Manoteras, 32 28050 Madrid
splatasb@grupogesfor.com

Ramón Alcarria
Depto Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid
Ciudad Universitaria s/n 28040 Madrid
ralcarria@dit.upm.es

***Resumen*—Este artículo presenta una plataforma OSGi para el despliegue de Servicios hospitalarios. Cardea es una plataforma de servicios de seguimiento hospitalario basada en estándares que permite el despliegue de servicios de forma estándar, segura y controlada, facilitando la integración de la nueva generación de servicios multimedia de una forma homogénea. Se presenta la arquitectura general del sistema y se describen los principales servicios desarrollados. Estos servicios ofrecen funcionalidades tales como la localización e identificación de personas y medicamentos mediante tecnologías de radiofrecuencia, integración con sistemas externos utilizando tecnologías de bus, servicios colaborativos con acceso multidispositivio basados en SIP o la gestión de la información dinámica de las entidades modeladas en el sistema. Finalmente se muestran los resultados del piloto desplegado en el servicio de farmacia del Hospital Gregorio Marañón para la gestión de medicamentos dentro del recinto de farmacia.**

***Palabras Clave*—RFID, OSGi, ESB, hospital, contexto**

I. Introducción

Un recinto hospitalario es un entorno complejo y dinámico con un enorme número de habitaciones y dependencias, si bien no son caóticos, tienen un alto grado de impredecibilidad, dado el continuo cambio de personal sanitario, pacientes y medicamentos. La farmacia hospitalaria es uno de los servicios con mayores necesidades de mejora en la automatización, control y seguimiento de medicamentos. La seguridad y el control de la trazabilidad es fundamental para los casos de retirada de lotes defectuosos o medicamentos no autorizados. Los sistemas actuales de gestión de medicamentos basados en lecturas de códigos de barras inducen a errores en la gestión del inventario. Estos errores viene provocados por una gestión manual de la entradas y salidas de medicamentos por parte del personal farmacéutico a través de los sistemas tradicionales de lectura de códigos de barras, pudiendo haber salidas y entradas de medicamentos no registradas. Por otro lado, el control manual del estado de los medicamentos puede provocar falta de previsión de bajo stock o vencimiento de la caducidad de medicamentos, provocando ineficiencias en el servicio y un desaprovechamiento de recursos. Además los sistemas actuales no utilizan de forma automática la información ambiental de las personas y activos, como puede ser su localización dentro del recinto, información de temperatura y humedad, o información proveniente de sensores biométricos.

En la línea de trabajo de dar solución a los problemas detectados en el entorno sanitario, Cardea ha propuesto una plataforma de servicios hospitalarios dentro del Subprograma Avanza I+D 2008 (TSI-020302-2008-78), que aborda el problema del desarrollo de una plataforma de servicios abierta y estándar basada en tecnologías OSGi, RFID y SIP.

Por una parte, se ha definido una plataforma que permite el despliegue de servicios de forma estándar, segura y controlada. El framework OSGi[2] permite la definición de unas especificaciones estándar para el desarrollo de servicios. Esto permite el desarrollo de nuevos servicios por parte de terceros y una gestión integral y remota de los mismo, creando un entorno escalable y configurable en función de las necesidades específicas del entorno de despliegue.

Por otra parte, se han definido y desplegado unos servicios básico que permiten cubrir las necesidades iniciales de la plataforma, integrando tecnologías que cubren los requisitos iniciales de la plataforma. Las tecnologías de radiofrecuencia RFID [10] han demostrado su potencial para la identificación de personas y objetos, por lo que su uso se está extendiendo en entornos complejos dónde la trazabilidad e inventariado son importantes. Por otro lado, el uso de SIP combinado con OSGi permite el desarrollo de servicios colaborativos entre los distintos actores del entorno hospitalario. Además, la gestión de información contextual permite el acceso a información dinámica y estática de las entidades a gestionar, permitiendo controlar la trazabilidad o detectar posibles alertas.

El resto del artículo se estructura como sigue. La sección II enmarca el escenario inicial de la plataforma. La sección III describe la arquitectura de la plataforma propuesta, describiendo con detalle cada uno de sus componentes. La sección IV describe la integración de la pila SIP en la plataforma y en la sección V se detalla la integración de la plataforma con servicios de empresa. Por último, se describe un piloto implantado en el Hospital Gregorio Marañón en la sección VI y se recogen las conclusiones y trabajos futuros sección VII.

II. Escenario inicial de Cardea

Cardea pretendía ofrecer una plataforma de servicios para ser desplegada en un servicio hospitalario. El servicio elegido fue el servicio de farmacia en el hospital Gregorio Marañón. Este servicio se encarga de recibir los pedidos de los medicamentos por parte de los proveedores y de elaborar medicamentos en los laboratorios propios. Una vez recibidos, los medicamentos se clasifican y se almacenan de forma adecuada en las dependencias desde las cuales el personal farmacéutico despacha los medicamentos a los pacientes con receta médica. Además de despachar los medicamentos por ventanilla, también se distribuyen medicamentos a las

diferentes áreas del Hospital Gregorio Marañón. Cardea pretendía satisfacer las carencias del sistema actual basado en lectura de código de barras, el cual inducía muchos errores humanos a la hora de gestionar el inventario de los medicamentos.

Los requisitos iniciales de la plataforma fueron:

- El sistema debía localizar e identificar al personal farmacéutico, pacientes y medicamentos dentro de las dependencias del servicio de farmacia.
- El sistema debía gestionar la diferente información dinámica y estática de las diferentes actores del sistema. El sistema debía gestionar el estado de los medicamentos, quién los ha dispensado y a qué paciente.
- El sistema debía proporcionar servicios aviso al personal farmacéutico en situaciones de alerta.
- El sistema debía permitir la sincronización de la información relevante con los diferentes sistemas utilizados actualmente, como el sistema de gestión de los medicamentos.

Partiendo de estos requisitos, en la siguiente sección se muestra la arquitectura del sistema.

III. Arquitectura de Cardea

La arquitectura del sistema Cardea [6] se muestra en la figura 1. En esta arquitectura se distinguen tres componentes principales: la plataforma de despliegue de servicios, los servicios desarrollados y la infraestructura RFID.

El núcleo principal de la plataforma de servicios Cardea es el **framework OSGi** (Open Service Gateway initiative). Este framework proporciona un entorno estandarizado para las aplicaciones (conocidas como bundles), definiendo un modelo de ciclo de vida para ellas y un registro de servicios (service registry) para facilitar la integración y descubrimiento entre aplicaciones, generando un modelo de componentes dinámico y completo. Entre los múltiples frameworks existentes, el framework Knopflerfish [9] ha sido elegido por proporcionar la funcionalidad requerida para la plataforma, destacando la consola de administración que permite un control remoto de las aplicaciones desplegadas. Sobre este framework OSGi se despliegan los diferentes servicios hospitalarios desarrollados.

La localización e identificación se realiza mediante el uso de etiquetas, lectores y antenas **RFID**. Dadas las especificaciones de trazabilidad requeridas para este sistema como distancia, velocidad de lectura y licencias, se eligió la banda UHF para realizar la identificación. Las etiquetas permiten identificar a los diferentes elementos trazables a los cuales son adheridas. El registro de estas etiquetas se realiza mediante un lector de tarjetas integrado en una PDA. Esta PDA contiene una aplicación de lectura, actualización y registro de etiquetas, que permite actualizar la información en la plataforma Cardea. La identificación de la localización se consigue mediante la lectura de las etiquetas por parte de arcos de antenas situadas en los puntos de control. Estas antenas están conectadas a concentradores que envían la actividad a través de la red hacia la plataforma.

La integración con sistemas RFID no está cubierto por OSGi. Soluciones previas han propuesto la integración de estos sensores en el framework [12], sin embargo en Cardea se integra el sistema de captura de información de los sensores

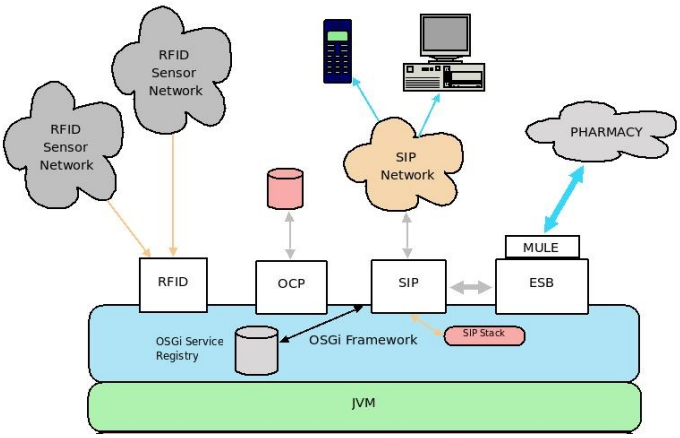


Fig. 1. Arquitectura de Cardea

RFID con aplicaciones que hacen uso de esta información. El **bundle RFID** permite manejar e interpretar la información enviada a través de la red por parte de la infraestructura RFID.

Cardea proporciona una plataforma de servicios hospitalarios sensible al contexto. Contexto hace referencia a los factores ambientales y circustanciales que proporcionan información relevante para las diferentes personas y elementos involucrados en el sistema, tales como situación, temperatura y humedad o dispensación. Los actores involucrados en el sistema, como el personal hospitalario y los medicamentos, componen un conjunto de elementos heterogéneos. Se han definido ontologías para su representación, ya que su uso no requiere una correspondencia exacta entre la información disponible y la requerida por el sistema [4][5][8][1]. Se ha integrado una ontología basada en la Web Semántica y en el estándar médico HL7, gestionada a través de Jena [7], un API de reglas del conocimiento y un middleware para capturar, almacenar y proveer información de contexto. Con ello, se ha conseguido un procesamiento inteligente de dicha información contextual que permite generar proactividad en la aplicación. El subsistema descrito recibe el nombre de **OCP** (Open Context Platform).

La plataforma Cardea debe permitir la comunicación con dispositivos y sistemas ajenos a la plataforma. OSGi proporciona servicios que permiten comunicarse con el exterior usando el protocolo http. Estos servicios no proporcionan la flexibilidad para integrarse con dispositivos y sistemas externos. Cardea proporciona un servicio que permite el establecimiento de sesiones SIP con dispositivos que soporten este protocolo. El **bundle SIP** permite a las aplicaciones desplegadas en Cardea el envío de mensajes de alarmas hacia los agentes SIP integrados en los diferentes dispositivos, proporcionando así un acceso multidispositivo a la plataforma. La arquitectura de este bundle se mostrará en la sección IV. Por otra parte, para cubrir la integración con sistemas externos, Cardea proporciona un middleware de mensajería síncrona y asíncrona que proporciona esta comunicación. El **bundle ESB** proporciona la flexibilidad para la definición los puntos de acceso a los servicios externos, así como la capacidad de lógica y de transformación de la información recibida para adecuarla a los sistemas externos. La integración con servicios de empresa se detalla en la sección V.

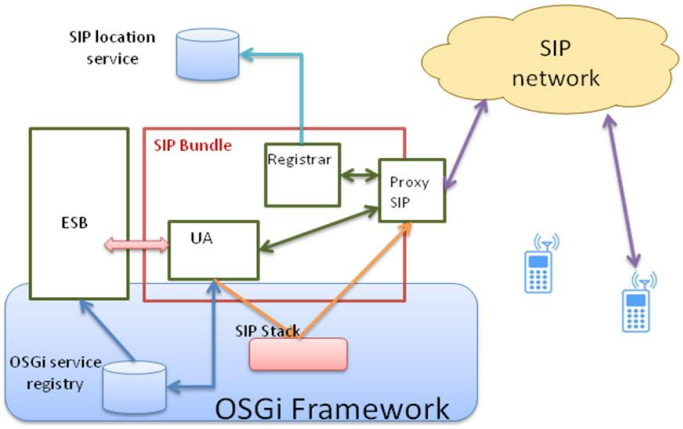


Fig. 2. Arquitectura del bundle SIP

IV. Servicios SIP

La integración de servicios SIP permite establecer comunicación entre el framework de OSGi y elementos externos utilizados para la recepción de mensajes de alarma. La arquitectura del servicio se muestra en la Fig. 2. Este servicio integra principalmente tres elementos de la arquitectura SIP:

- **Agente de usuario**: Constituye el elemento principal de las comunicaciones SIP. El protocolo SIP es un protocolo de nivel de aplicación que utiliza la estructura cliente-servidor para realizar conexiones extremo a extremo. Los elementos situados en los extremos son los agentes de usuario (UA).
- **Servidor proxy**: Un proxy es un elemento intermedio del proceso de comunicación SIP que se encarga de recibir peticiones SIP y reenviarlas hacia su destino. Aunque no es un elemento fundamental para desarrollar un sistema de comunicaciones, se ha añadido a la arquitectura para observar el comportamiento de la red SIP.
- **Servidor de registro**: El servidor de registro es un elemento encargado de recibir las peticiones de registro SIP y su procesamiento, actualizando la información del agente de usuario (dirección, disponibilidad u otras preferencias) situado en el servicio de localización (SIP Location Service). El servicio de localización se implementa como una tabla "Hashtable" donde se almacena la correspondencia entre la dirección URI de los usuarios y su dirección física.

La integración de SIP en un sistema OSGi genera dos ventajas fundamentales. En primer lugar se convierte cualquier servicio OSGi en un SIP UA "virtual". De esta forma, un dispositivo SIP externo puede establecer comunicaciones utilizando este protocolo sobre un entorno de desarrollo OSGi. Por otra parte, se extiende OSGi para poder manejar un dispositivo SIP desde las aplicaciones instaladas en su framework. De esta forma se dota a los servicios desplegados en la plataforma OSGi de las características de movilidad, presencia, flexibilidad del protocolo SIP.

Cardea permite que los terminales móviles o PDAs, con capacidad de comunicaciones por SIP, se registren en el servidor de registro implementado en la plataforma. En este momento, las aplicaciones desplegadas en Cardea pueden hacer uso de la funcionalidad proporcionada por el bundle SIP, permitiendo conectarse al dispositivo y enviarle mensajes usando la URI registrada.

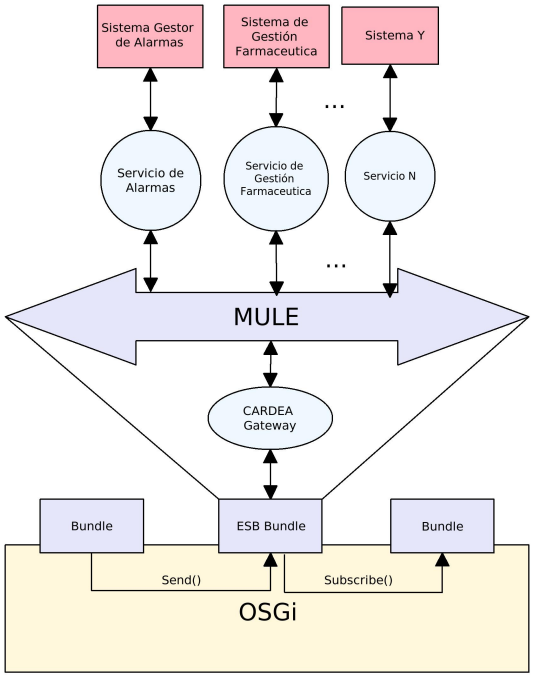


Fig. 3. Arquitectura ESB

V. Servicios de Empresa

Gracias a la gran aceptación de los ESB (Enterprise Service Bus) en la actualidad, podemos encontrar conectores con los principales productos de código abierto y componentes COTS (Commercial Off The Shelf). El empleo de servicios web, tanto REST como SOAP reduce drásticamente las tareas de integración, que se desacoplan gracias a la integración en el bus.

El ESB [3] es un elemento middleware que se basa en mensajería síncrona o asíncrona y que proporciona interoperabilidad segura entre aplicaciones de empresa por medio de XML, interfaces de Servicios Web y reglas de enrutamiento estandarizado de documentos. En la práctica esto significa que los datos son transferidos hacia y desde sus destinatarios basados en directrices preestablecidas que son comunes a todos los grupos que comparten la información, para asegurar que los datos se mantienen tal y como son enrutados.

Cardea integra en su plataforma de despliegue un bundle ESB. Este bundle proporciona servicios de envío y suscripción de mensajería al resto de bundles desplegados en la plataforma. El bundle usa el framework de mensajería Mule [11] que permite un configuración dinámica de puntos de entrada, puntos de salida y colas.

El bundle ESB despliega un bus que interconecta los servicios de empresa que dan soporte a las diferentes aplicaciones con las que la plataforma Cardea interacciona (Fig. 3). Los servicios de mensajería se definen mediante un fichero XML de configuración del bus, en el cual se establecen los diferentes servicios, sus puntos de entrada y salida al bus, así como componentes y transformadores de mensajes. De esta forma, la configuración del sistema se hace de manera

rápida y dinámica.

Se ha establecido un servicio denominado Cardea Gateway, el cual es el encargado de enviar al bus los datos procedentes de la plataforma hacia otras aplicaciones, así como de obtener del bus diferentes datos y redirigirlos a los correspondientes servicios de la plataforma. Cada uno de los servicios tiene asociados una serie de transformadores desarrollados en java encargados de adecuar los datos que se introducen en el bus al formato adecuado esperado por las aplicaciones destino.

Con todo ello, el servicio ESB desplegado en Cardea es altamente escalable, permitiendo la integración con nuevos sistemas de empresa de forma dinámica mediante la definición de la configuración. Maneja todas las interacciones entre aplicaciones y componentes de forma transparente, independientemente si las aplicaciones están desplegadas en la misma máquina o si son accesibles a través de internet, y también es independiente de los protocolos de transporte utilizados.

VI. Experimentación

El sistema propuesto ha sido implantado de forma experimental en el servicio de farmacia del Hospital Gregorio Marañón durante el mes de marzo de 2009. La plataforma Cardea pretendía gestionar un grupo de ocho medicamentos representativos que permitiese evaluar la plataforma. Durante la fase de experimentación, sólo los medicamentos fueron gestionados a través de la plataforma Cardea, ya que la involucración de pacientes hubiese requerido la autorización de los mismos debido al manejo de información sensible de los pacientes y ésto hubiese provocado retrasos en la implantación.

En este escenario de experimentación se definieron dos servicios externos. Estos servicios fueron simulados para que la plataforma no fuera intrusiva con los sistemas actuales. Los sistemas definidos fueron:

- **Sistema Gestor de Alarmas**: Este sistema se encarga de la gestión de las alarmas producidas. En Cardea, se definieron dos tipos de alarmas: un medicamento está caducado o el stock de un tipo de medicamento está por debajo del umbral deseado. El bundle OCP es el encargado de detectar las alarmas por caducidad y el bundle RFID de la gestión de stock bajo. Estos bundles hacen uso de los servicios del bundle ESB para enviar las alarmas a este sistema de gestión de alarmas.
- **Sistema de Gestión Farmacéutica**: Este sistema es el encargado de gestionar la actividad de los medicamentos dentro de la farmacia. El bundle OCP detecta los cambios de contexto de los medicamentos y envía la información de contexto hacia el sistema de gestión farmacéutica usando los servicios del bundle ESB.

Estos sistemas fueron simulados mediante la exposición de servicios web que recogen la información del bus y una aplicación web que mostraba los diferentes mensajes de alarma y actividad, permitiendo evaluar el correcto funcionamiento de la plataforma.

Por otro lado, las alarmas detectadas dentro de la plataforma eran enviadas a su vez a un agente SIP externo a través de los servicios ofrecidos por el bundle SIP.

La tasa de etiquetas no leídas de forma automática por la red de antenas ha sido inferior al 2.5%. Los principales factores identificados de esta tasa son:

- Registro erróneo por parte del personal farmacéutico al adherir las etiquetas a los envases.
- Velocidad de lectura. Durante la entrada masiva de medicamentos, el tiempo de paso por los arcos de antena no era suficiente para una lectura correcta de todas las etiquetas.

VII. Conclusiones y Futuros Trabajos

En este trabajo de investigación se ha presentado una plataforma basada en tecnologías OSGi, SIP y RFID para la localización e identificación de medicinas en un entorno hospitalario. El trabajo ha utilizado un framework OSGi que facilita el despliegue de los diferentes módulos utilizados y permite el despliegue de nuevos servicios. La plataforma expuesta, aunque se ha modelado para entornos hospitarios, puede fácilmente reutilizarse para otros entornos generando una nueva ontología para el nuevo entorno y definiendo nuevos servicios de empresa

Finalmente se ha presentado un piloto desplegado en el servicio de farmacia del Hospital Gregorio Marañón durante el periodo de funcionamiento del piloto, los datos recogidos del sistema fueron muy satisfactorios tanto para el personal técnico encargado del despliegue y mantenimiento del sistema, cómo del personal sanitario involucrado.

El uso de la plataforma permite la posibilidad de utilizar esta arquitectura para otros entornos, así como ofrecer nuevas líneas de investigación dentro del entorno hospitalario, como puede ser la optimización y mejora de la calidad de los servicios hospitalarios al personal sanitario, pacientes y familiares, así como otro tipo de activos del hospital

Agradecimientos

Cardea ha sido cofinanciado por el Ministerio de Industria, Turismo y Comercio, dentro Subprograma Avanza I+D 2008 (TSI-020302-2008-78)

Referencias

[1] Gu, T.; Wang, X. H.; Pung, H. K. and Zhang, D. Q. An ontology-based context model in intelligent environments, 2004.

[2] O. Alliance. *OSGi Service Platform Core Specification*, 2007. Más información en http://www.osgi.org/.

[3] V. Cabrera. Entreprise Service Bus, 2006.

[4] H. Chen, T. Finin, and A. Joshi. An ontology for context-aware pervasive computing environments. In *Workshop on Ontologies and Distributed Systems*. IJCAI-2003, August 2003.

[5] H. Chen, F. Perich, T. W. Finin, and A. Joshi. Soupa: Standard ontology for ubiquitous and pervasive applications. In *MobiQuitous*, pages 258–267. IEEE Computer Society, 2004.

[6] I. Informática. Sitio Web Cardea. Disponible en http://cardea.germinus.com/.

[7] JENA. Jena semantic web framework. Disponible en http://jena.sourceforge.net/.

[8] M. Klein, A. Schmidt, and R. Lauer. Ontology-centred design of an ambient middleware for assisted living: The case of soprano. In T. Kirste, B. König-Ries, and R. Salomon, editors, *Towards Ambient Intelligence: Methods for Cooperating Ensembles in Ubiquitous Environments (AIMCU), 30th Annual German Conference on Artificial Intelligence (KI 2007), Osnabrück, September 10, 2007*, 2007.

[9] Knopflerfish. Knopflerfish OSGi Framework. Disponible en http://www.knopflerfish.org/.

[10] M. Meints. D3.7 A Structured Collection on Information and Literature on Technological and Usability Aspects of Radio Frequency Identification (RFID). FIDIS deliverable, Junio 2007.

[11] M. Source. The open source choice for integration and SOA. Disponible en http://www.mulesource.com/.

[12] J. Wu, D. Wang, and H. Sheng. Design an OSGi Extension Service for Mobile RFID Applications. *E-Business Engineering, IEEE International Conference on*, 0:323–326, 2007.

Generación de tráfico de ataque para la evaluación de sistemas de detección de intrusos

R. Salazar-Hernández, J. Díaz-Verdejo
Dpto. de Teoría de Señal, Telemática y Comunicaciones – Centro de Investigación en TIC (CITIC- UGR)
Universidad de Granada
c/Daniel Saucedo Aranda s/n 18071 Granada
[rsalaza,jedv]@ugr.es

***Resumen*- Los sistemas de detección de intrusos basados en red y detección de anomalías necesitan utilizar tráfico de red para realizar las fases de entrenamiento, prueba y validación. Este tráfico debe contener patrones de comportamiento normal y anómalo y representar adecuadamente el tráfico real. Sin embargo, no es fácil obtener un conjunto representativo de los ataques existentes. En este artículo se describen varias aproximaciones para obtener tráfico de ataques correspondientes al protocolo HTTP. Se han obtenido de varias fuentes la información los *exploits* necesarios para generar los ataques dentro de un entorno controlado. Las bases de datos así recopiladas han sido sometidas a evaluación con un sistema de detección de intrusos basado en red para analizar su comportamiento y calidad.**

***Palabras Clave*-** IDS, ataques, *exploits*, seguridad de web

I. INTRODUCCIÓN

Dentro de la gama de servicios que ofrece Internet resulta patente que el servicio World Wide Web (web) es uno de los de mayor uso [1]. En consecuencia, se están incrementando significativamente los niveles de dependencia de la sociedad respecto de la disponibilidad de este servicio. Sin embargo, como cualquier otro servicio en Internet, el servicio web puede ser vulnerable a ataques de todo tipo y por muy diversos procedimientos. En este contexto, los cortafuegos, los sistemas de detección de intrusos y otros similares representan la principal línea de defensa perimetral, esto es, en cuanto al acceso externo al servicio.

Los sistemas de detección de intrusos (IDS, del inglés *Intrusion Detection Systems*) tienen como objetivo la detección de actividades intrusivas, esto es, de ataques, y la consiguiente emisión de alertas y/o activación de mecanismos de respuesta, en su caso [2]. Para ello se basan en la monitorización de eventos y su clasificación como actividad inocua o ataque. En función del tipo de eventos monitorizados, se suelen diferenciar los N-IDS (del inglés *Network-based IDS*) y los H-IDS (del inglés *Host-based IDS*). En el primer caso, el IDS monitoriza el tráfico de red, mientras que en el segundo se monitorizan los eventos internos del sistema. Por otra parte, también se suelen diferenciar dos tipos básicos en función de la metodología para la detección de la actividad intrusiva. Así, se diferencian los basados en la detección de anomalías (A-IDS, del inglés *Anomaly-based IDS*) y los basados en la detección de firmas (S-IDS, del inglés *Signature-based IDS*) [3].

El presente trabajo se centra en el desarrollo de sistemas de detección de anomalías sobre tráfico de red (ANIDS, *Anomaly-based Network-based IDS*). Para ello es necesario contar con tráfico de red convenientemente etiquetado como tráfico normal o de ataque, a fin de realizar el entrenamiento y la evaluación del sistema resultante. La capacidad del detector como clasificador queda determinada tanto por la capacidad de detectar los ataques (verdaderos positivos, TP) como por el número de eventos no intrusivos clasificados como ataques (falsos positivos, FP). Para una evaluación correcta de estos indicadores se requiere tráfico real capturado de una red en explotación [4] conveniente y fehacientemente etiquetado como normal o ataques. El tráfico normal debe ser suficiente como para obtener un modelo de normalidad adecuado, mientras que el tráfico de ataques debería incluir el mayor número de variantes, siendo difícil el cumplimiento de estos y otros requisitos en una captura de tráfico real [5].

En el presente trabajo se propone como alternativa el uso de tráfico real capturado y filtrado mediante un IDS para eliminar los ataques o, al menos, gran parte de ellos, y adquirir un conjunto extenso y completo de los ataques conocidos a partir de su generación artificial. En particular, el objetivo de este trabajo es determinar una metodología que permita generar y capturar tráfico de ataques con el menor esfuerzo posible. El estudio se ha centrado en ataques contenidos en las URI de peticiones GET del servicio HTTP, por ser éstos objeto de investigación en nuestro grupo de trabajo.

El resto del artículo está organizado como se describe a continuación. En la Sección II, se describe el escenario utilizado. Las aproximaciones propuestas para la recopilación de los ataques se exponen en la Sección III. En la Sección IV se presenta la clasificación de los ataques de acuerdo a varias taxonomías. Las estadísticas resultantes de la evaluación se presentan en la Sección V. Finalmente, en la Sección VI se presentan algunas conclusiones del presente trabajo, así como las líneas futuras.

II. ESCENARIO EXPERIMENTAL

Los objetivos planteados aconsejan el establecimiento de un escenario de experimentación que reúna las características adecuadas. En particular, resulta relevante que el entorno esté controlado en todo momento y que se encuentre aislado a fin de evitar interferencias de otros sistemas. Por otra parte, se requiere de la existencia de diferentes equipos que desempeñarán el papel de atacantes o víctimas. En este

	Paquetes		Alertas	
Detector	Total	GET	Únicas	Total
wikto	10944	2157	237	491
Nikto (ev.7)	12176	2085	102	462
Nikto (ev.8)	15987	2706	312	983
W3af	14127	2678	317	975
Nstealth	109659	10977	482	4711
Total	162893	20603	1450	7622

Tabla 1. Tráfico generado mediante buscadores de vulnerabilidades y clasificado mediante Snort.

contexto, en [6] y [7] se utiliza una infraestructura basada en maquinas virtuales con la finalidad de aislar el sistema experimental de entornos reales. Análogamente, para prevenir los posibles daños y aislar la red de trabajo, se propone la utilización de máquinas virtuales para simular un grupo de equipos con diferentes sistemas operativos.

El escenario propuesto se compone de varios equipos, todos simulados mediante *VMare Workstation* [8], y dispuestos dentro de un mismo segmento de red. Los equipos virtuales podrán desempeñar dos roles: víctimas o atacantes. Se ha dispuesto un único equipo víctima junto con varios equipos atacantes. Al equipo víctima se le ha asignado también el papel de monitor, habilitándose como sensor para capturar todo el tráfico de red. Los ataques son ejecutados lanzándolos desde las máquinas atacantes hacia el equipo víctima.

Este escenario será utilizado tanto para capturar como para evaluar los resultados de su ejecución. Así, en primer lugar se generarán varias baterías de ataques, mediante los programas o técnicas adecuados, que serán capturados por el sensor (*Wireshark* [9] en este caso). Idealmente, una batería de ataques será, por tanto, la recopilación de URI generados durante la fase de recopilación de ataques.

Una vez dispuestas las baterías de ataques, se procederá a evaluar su comportamiento. El modo de evaluación, se utilizará *Snort* [10] como sensor y clasificador del tráfico. En este modo, sólo se considerarán dos equipos: un atacante basado en *Linux Backtrack* [11] y una víctima.

III. Recopilación de ataques

En esta sección se presentan y discuten los métodos evaluados para la recopilación de tráfico de ataques.

A. *Generación de ataques mediante buscadores de vulnerabilidades*

En una primera aproximación se evaluará el uso de detectores de vulnerabilidades para la generación de tráfico de ataques. Esta aproximación se describe en [12] y [13], donde se utilizan buscadores de vulnerabilidades como *Nessus* [14] a fin generar tráfico de ataques para evaluar los IDS. En nuestro caso, se procede a utilizar algunos detectores como *nikto* [15], *wikto* [16], *w3af* [17] y Nstealth [18], para generar tráfico.

La evaluación de la base de datos obtenida mediante *Snort* muestra que se genera tráfico adicional durante la búsqueda de la vulnerabilidad, resultando, en consecuencia, la existencia de tráfico normal entremezclado con el de ataques. En la Tabla 1 se muestran los resultados de la captura y su clasificación mediante Snort (reglas *uricontent* en VRT de 28/12/08). La columna "Únicas" corresponde a los tipos de ataque detectados, mientras que la "Total" representa el total de instancias de ataques detectados. A este respecto, es importante notar que un mismo ataque puede presentar varias instancias resultantes de alguna pequeña variación de los parámetros del mismo. Como se puede observar, no todo el tráfico capturado (tráfico GET) es etiquetado como ataque, siendo significativas las discrepancias.

Reglas Snort por referencia		Núm. de firmas
URL		347
Sin referencia		234
Bugtraq		553
Nessus		45
Common Vulnerabilities and Exposures (CVE)		73
ARACHNIDS		61
	Bugtraq	49
	CVE	3
	URL	1
	Sin referencia	7

Tabla 2. Distribución de las firmas de Snort según la referencia.

En consecuencia, en el tráfico generado por los detectores de vulnerabilidades existe tráfico del cual no tenemos la certeza de que sea tráfico de ataques. Aunque los resultados de esta primera aproximación son buenos, en cuanto a la cantidad del tráfico de ataques generado y la relativa facilidad de su obtención, no son adecuados para evaluar la actuación de los sistemas de detección de intrusos.

B. *Uso de conjuntos de reglas de Snort*

Como aproximación alternativa se ha considerado como origen de la información el conjunto de reglas de Snort asociado a las URI. A partir del propio conjunto de reglas de Snort, se deberían poder generar todos los ataques potencialmente detectables por dicho IDS. Para la generación de los ataques será necesario, en función de su naturaleza, reproducir el URI, si no necesita ningún tipo de parametrización, o localizar un *exploit* que lo genere. Por tanto, esta aproximación presenta el inconveniente de requerir la recopilación manual de los *exploits* necesarios. Por el contrario, por construcción, debe presentar una mayor cobertura de los tipos de ataque conocidos.

Se han considerado las reglas VRT del 16 de agosto del 2007, que incluyen 2,375 reglas que afectan al protocolo HTTP (caso de estudio) de un total de 8262 reglas. El análisis de las firmas resultantes en función del origen de la referencia proporciona los datos mostrados en la Tabla 2. La fila URL corresponde a las firmas asociadas a un URL fijo que, por tanto, no necesita parametrización y para el que no es necesario localizar un *exploit*. La fila "Sin referencia" muestra el número de reglas que no presentaban el campo "*reference*", lo cual significa que son reglas aportadas por el personal certificado de Snort. Éstas no serán consideradas en lo sucesivo. Las reglas con referencia de Arachnids tuvieron que ser revisadas debido a que el portal asociado, www.whitehats.com, ha sido dado de baja. Las 61 firmas incluidas, tras su revisión, fueron redistribuidas entre las restantes fuentes como se muestra en la Tabla 2. Las vulnerabilidades descritas en Bugtraq (www.securityfocus.com), Nessus (www.nessus.org) y CVE

Fuente	Total paq.	Paq. ataque
ArachNIDS	1,345	96
Nessus	916	73
Bugtraq	38,120	953
CVE	817	59
Total	41,198	1,181

Tabla 3: Tráfico de ataques recopilados mediante el uso de exploits.

OSVDB / OSWAP	6 Input Manip.	8 DoS	10 Inform. disclosure	12 Auth. Manag.	18 Lost Integrity	29 Web Related	TOTAL
1 Absolute path traversal	6/8		34/66	24/55		6/11	2,25/2,25
2 Full path disclosure	2/5		3/8	4/12			9/25
3Account lockout attack	3/5			3/11			6/0,87
4 Path manipulation	5/5		4/16	1/1	1/2		2,75/2,75
5 Relative path manipulation			10/19	19/28		1/1	2,2/2,2
6 Forced browsing	7/12		2/2	7/13		3/7	2,55/2,55
7 Denial of service		18/33		1/1			1,55/1,55
8 XSS using script in attributes	4/4		4/7	12/28			2/2
9 XSS cross-site scripting	1/1			2/5	1/1		4/2,4
10 Buffer overflow attack	2/4		1/1	13/22			2,09/2,09
11 Command injection	6/6		50/122	43/88		3/7	2,33/2,33
12 Resource Injection			2/2	1/2			3/1,5
13 Double encoding	5/5		6/16	5/39			1,5/1,5
14 Setting manipulation	3/5		3/3			2/3	8/2,27
15 SQL injection			3/7	2/8			5/0,68
Total	8,43/8,43	0,55/0,55	7,45/7,45	7,05/7,05	1,5/1,5	3,07/3,07	6/6

Tabla 4: Clasificación de los ataques según taxonomías OSVDB y OSWAP. Se indican, con el formato A/B el número de ataques (A) y de instancias (B).

(cve.mitre.org) fueron consideradas para la recopilación de los *exploits* correspondientes.

A partir de esta información y, en su caso, de los *exploits* recopilados de cada fuente, se lanza una batería de ataques de la que resultan 1181 paquetes GET (Tabla 3). El análisis de esta batería de ataques mediante Snort usando el mismo conjunto de reglas considerado como punto de partida proporciona únicamente 957 alertas. Se constata, en consecuencia, un comportamiento inconsistente en esta aproximación, poniéndose en duda la calidad de las reglas y de los *exploits* recopilados. En particular, una inspección somera de los *exploits* muestra que muchos de ellos tienen una finalidad descriptiva de la naturaleza del ataque, no estando parametrizados ni generando instancias reales del mismo, sino plantillas que muestran la estructura de los paquetes, tal como se muestra en el apartado siguiente.

C. *Recopilación y generación supervisada a partir de bases de datos de vulnerabilidades y exploits*

En Internet existen múltiples fuentes de información respecto de ataques y vulnerabilidades. Entre ellas, resultan relevantes *Bugtraq* de *Securityfocus* [19] y *Open Source Vulnerabilities Data Base* (OSVDB) [20], en las que se han encontrado claramente documentadas las vulnerabilidades que afectan al protocolo HTTP así como referencias a los correspondientes *exploits*. De las fuentes de colección de ataques, la base de datos Bugtraq de Securityfocus cuenta con una descripción más explícita de la vulnerabilidad, por lo que se ha decidido tomar esta fuente de información como punto de partida para la realización de esta etapa.

En cualquier caso, respecto de los objetivos del presente trabajo, se han encontrado los problemas que a continuación se detallan:

a) Parametrización del ataque

Se han encontrado casos en los que el *exploit* correspondiente a la vulnerabilidad no presenta una adecuada parametrización. Sin embargo, en otros casos se encuentra claramente el URL que explota una vulnerabilidad.

Este problema ya había sido observado en las aproximaciones previamente realizadas. En este caso se va a proceder a la revisión manual de los mismos para su adecuación.

b) Búsqueda de los exploits

En otros casos no se indica el URI ni su formato, siendo necesario localizar el *exploit* adecuado a partir de las referencias proporcionadas. Para ello se ha realizado una búsqueda de los *exploits* en las diversas fuentes disponibles en Internet. Obviamente, se ha tomado como punto de partida la información contenida en *Bugtraq*. Siguiendo el flujo mostrado, si el *exploit* es encontrado se parametriza, adecuándolo según el lenguaje de programación en el que se encuentre. Una vez realizada esta tarea, y haciendo uso del escenario descrito previamente, se ejecuta el *exploit*, capturándose y almacenándose las trazas correspondientes. Cuando el *exploit* no sea encontrado se continúa la búsqueda en otras fuentes de información dispuestas en internet. Como último recurso, la búsqueda del ataque se realizo en foros de discusión, listas de correos o mediante una búsqueda general en Internet.

Mediante el método descrito se han encontrado ataques muy variados y con argumentos varios para ser ejecutados. Se han recopilado 338 ataques con 707 instancias, que han sido almacenados en la base de datos denominada RDB.

IV. CLASIFICACIÓN DE LOS ATAQUES

A fin de determinar la respuesta ante ataques de diversa naturaleza, lo que puede facilitar la incorporación e mejoras y el análisis de resultados, puede resultar conveniente que el conjunto de ataques utilizado se encuentre categorizado en función del tipo al que pertenecen.

A este fin se han considerado dos taxonomías disponibles en Internet: OSVDB [20], que incluye una clasificación de todas las vulnerabilidades existentes debidamente actualizada; y *Open Web Application Security Project* (OWASP) [21]], que realiza una clasificación únicamente de los ataques a las aplicaciones Web. Ambas taxonomías son complementarias.

En consecuencia, cada uno de los ataques contenidos en la base de datos RDB ha sido categorizado atendiendo a ambas taxonomías, obteniéndose los resultados mostrados en la Tabla 4. Se puede observar que el mayor número de ataques recopilados, así como las instancias de estos, corresponden a los de gestión de autenticación, de acuerdo a OSVDB, con un total de 137 ataques con 313 instancias de ataques. En

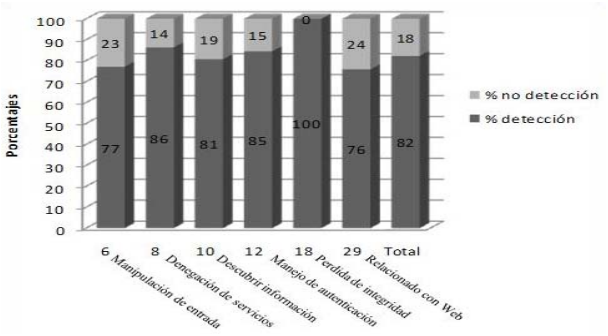


Fig. 1: Tasas de detección por categorías (OSVDB) utilizando Snort.

contraste, la clasificación usando OSWAP muestra un notorio cambio en el que la mayor representación se encuentra en la inyección de comandos (*command injections*), con un total de 102 ataques y 223 instancias de ataques.

V. EXPERIMENTACIÓN

La base de datos recopilada (RDB) ha sido contrastada mediante Snort, a fin de determinar sus características y potencialidades respecto del proceso de detección de intrusiones. Los resultados obtenidos constituirán el sistema de referencia respecto al que se compararán los desarrollos de sistemas basados en anomalías que se realicen en nuestro equipo de investigación.

Para la etapa de experimentación se han tomado las reglas VRT del 23 de diciembre del 2008. Una de las características de este tipo de reglas (reglas VRT) es la verificación y validación de realizada por los expertos de Sourcefire.

El conjunto de reglas considerado contiene 9871 reglas, que han sido preprocesadas para seleccionar sólo las que afectan el protocolo http, resultando un total de 2635 reglas. De éstas, se han elegido y considerado únicamente las que afectan al contenido del URI. Finalmente, el conjunto de reglas utilizado es de 1620 reglas.

Se ha configurado Snort para que utilice este conjunto de reglas, deshabilitando los preprocesadores que pudieran alterar los URI. Se ha instalado y configurado el *front end* de Snort denominado *Basic Analysis and Security Engine* (BASE) [22] con el fin de facilitar el análisis de los datos.

En la Fig. 1 se muestran los ataques detectados y no detectados según su categoría. Como se puede observar, los ataques de pérdida de integridad son detectados con alta eficiencia (100%). Globalmente, se comprueba que Snort presenta un 17,83 por ciento de trazas de ataque no detectadas, por lo que su rendimiento ante esta batería de ataques puede calificarse como insuficiente.

Para mejorar la capacidad de detección se hace necesario mejorar la calidad de las reglas o desarrollar técnicas de detección alternativas.

VI. CONCLUSIONES

La obtención de tráfico real para la evaluación de los IDS resulta difícil tanto por cuestiones legales y de invasión de la privacidad, como por aspectos técnicos y del etiquetado del tráfico. En este artículo se han descrito varios métodos para obtener tráfico sintético con ataques al protocolo HTTP que permitan la evaluación de la actuación de los IDS basados en red. A pesar el planteamiento inicial relativo a la automatización de la recopilación de ataques, se ha mostrado la necesidad de supervisar los ataques recopilados. Debido a esto, resulta difícil conseguir una base de datos completa de los ataques conocidos. Por otra parte, la información disponible en las diversas fuentes consultadas llega, a veces, a resultar incongruente, dificultando aún más la recopilación de ataques con la certeza de que sean tales, aspecto fundamental para la caracterización de las capacidades de detección de los IDS.

Las estadísticas del tráfico de ataques recopilado muestran claramente que el rendimiento de los detectores basados en firmas es insuficiente para detectar todos los ataques, por lo que se requiere de técnicas adicionales que mejoren el rendimiento de los IDS. Para ello se pueden utilizar detectores basados en anomalías que pueden operar de forma conjunta con los basados en firmas (detectores híbridos), constituyendo éste un tema relevante de investigación.

Finalmente, las bases de datos de ataque recopiladas pueden ser utilizadas, dentro de las limitaciones de cada una de ellas, para evaluar el rendimiento de los IDS basados en anomalías y firmas.

AGRADECIMIENTOS

La participación de R. Salazar-Hernández ha sido posible gracias al programa PROMEP y a la UAT (México).

Este proyecto ha sido parcialmente financiado por el MICINN, a través del proyecto TEC2008-06663-C03-02.

REFERENCIAS

[1] A. Odlyzko, “Internet Traffic Growth: Sources and Implications”, *Proc. of SPIE*, vol. 5247, 2003, pp. 1-15.
[2] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, E. Vázquez, “Anomaly-based network intrusion detection: Techniques, systems and challenges”, Computers & Security, vol. 28, pp. 18-28, 2009.
[3] John McHugh, “Intrusion and intrusion detection”, International Journal of Information Security, vol. 1, pp. 14-35, 2001.
[4] McHugh, J.; “The 1998 Lincoln Laboratory IDS Evaluation. A critique.” *Proc. RAID 2000*, LNCS vol. 1907, pp. 145-161, 2000.
[5] M. Bermúdez-Edo, R. Salazar-Hernández, J. Díaz-Verdejo, and P. García-Teodoro, “Proposals on Assessment Environments for Anomaly-Based Network Intrusion Detection Systems”, *Proc. CRITIS 2006*, LNCS 4347, pp. 210 – 221, 2006.
[6] Massicotte, F.; Gagnon, F.; Labiche, Y.; Briand, L., Couture, M.; “Automatic Evaluation of Intrusion Detection Systems”, *Proc. ACSAC '06*.
[7] M. Laureano, C. Maziero, E. Jamhour; “Protecting host-based intrusion detectors through virtual machines”; Graduate Program in Applied Computer Science, Pontifical Catholic University of Paraná, Brazil. Available online 16 October 2006.
[8] VMware, Inc. http://www.vmware.com, 2006.
[9] Wireshark, http://www.wireshark.org
[10] M. Roesch, “Snort – lightweight intrusion detection for networks”, *Proc. 13th Conference on Systems Administration (LISA-99)*, 1999.
[11] Backtrack, http://www.remote-exploit.org, 2005
[12] Y. Wang, H. Abdel-Wahab, “A Correlative Context-based Framework for Network Intrusion Detection System”, *Proc. 10th ISCC 2005*, pp. 463-468, 2005
[13] Yamada, A.; Miyake, Y.; Takemori, K.; Tanaka, T.; “Intrusion detection system to detect variant attacks using learning algorithms with automatic generation of training data”, *Proc. ITCC 2005*, Vol. 1, pp. 650-655, 2005.
[14] Nessus, http://www.nessus.org, 2004.
[15] Nikto: Web Server Vulnerability Detection Tool, (http://www.cirt.net/code/nikto.shtml), 2006.
[16] Wikto; http://www.sensepost.com/research/wikto, 2004-2008.
[17] Andrés Riacho; “w3af, Web Application Attack and Audit Framework”, http://w3af.sourceforge.net/, 2008.
[18] Nstealth; www.nstalker.com/nstealth/
[19] Security Focus, Bugtraq; http://www.securityfocus.com; 1998-2009
[20] Jake Kouns, Chris Sullo, Brian Martin, David Shettler, Steve Torino, “Open Source Vulnerability Data Base”; http://osvdb.org; 2002-2009
[21] Open Web Application Security Project OWASP, http://www.owasp.org/
[22] BASE is the Basic Analysis and Security Engine; http://base.secureideas.net; 2004-2008.

Streaming P2P robusto en redes Ad-hoc utilizando información social

A.J. González, D. Rodríguez, J. López, F. I. Rillo, J. Alcober
Departamento de Ingeniería Telemática, Universitat Politècnica de Catalunya / Fundació i2cat
Av. Canal Olímpic 15, PMT-Edifi C4, Mediacat. Castelldefels
alberto.jose.gonzalez@upc.edu, dani.rodriguez@i2cat.net, javi.lopez@i2cat.net, rillo@eel.upc.edu, jesus.alcober@upc.edu

***Resumen*—En la actualidad, las herramientas colaborativas que ofrecen servicios basados en redes sociales, distribución de ficheros y *streaming* de contenidos multimedia (*Peer-to-peer*) entre comunidades virtuales de usuarios están teniendo un gran desarrollo e interés tanto en el ámbito académico como industrial. Actualmente existen numerosas aplicaciones de este tipo que son usadas a diario por millones de usuarios. Este hecho viene motivado principalmente por su capacidad de llegar a un gran número de usuarios de cualquier índole con un bajo coste de infraestructura, al no requerir un costoso servidor centralizado, y facilitando el trabajo colaborativo entre usuarios. No obstante, este tipo de entornos presentan ciertas limitaciones de operación dada su naturaleza dinámica y heterogénea. Estas operaciones se acentúan cuando nos centramos en un entorno móvil como es una red móvil Ad-hoc. En este trabajo presentamos una solución para realizar *streaming* P2P en una red móvil Ad-hoc, tomando el conocimiento estadístico de la red social formada por los usuarios de la aplicación, con el objetivo de mejorar el mecanismo de distribución del contenido basado en *Network Coding*.**

***Index Terms*—*Streaming* P2P, *Network Coding*, Redes Ad-hoc, Redes Sociales**

I. Introducción

El fenómeno de los servicios ofrecidos por las redes sociales ha dado un vuelco a la forma de interactuar en la población, llegando al punto en que estos servicios ofrecidos a través de Internet definen estructuras de comunidad social, como una organización o institución, con sus miembros y sus relaciones internas. Las relaciones sociales entre los miembros de una comunidad pueden ser de diferentes tipos: amistosa, cortés, profesional, académica, etc. [1].

Los servicios que se ofrecen en las redes sociales de Internet son numerosos, desde compartición de cualquier información (p.e. texto, fotos y vídeos), hasta creación de eventos en calendarios compartidos por las comunidades virtuales. Por otro lado, la propia demanda social generada por estos servicios ha hecho que los dispositivos móviles dispongan del mismo acceso[2] [3]. Las redes sociales presentan un marco adecuado para la compartición de cualquier tipo de información y, de manera natural, se ha implementado este tipo de servicio. Finalmente, existen servicios de compartición de archivos (en redes P2P) de los que han surgido redes sociales y viceversa. Incluso existen iniciativas para aprovechar el conocimiento del comportamiento de un usuario en el sistema P2P, para mejorar el rendimiento de los mecanismos de búsqueda de contenidos [4].

No obstante, una idea bastante atractiva consiste en implementar una solución de *streaming* de contenidos en entornos P2P sobre una red Ad-hoc, aprovechando la posibilidad de realizar *broadcast* de manera eficiente, robusta y escalable.

Sin embargo, las implementaciones tradicionales de P2P en redes móviles Ad-hoc tienen la problemática de que no son efectivas utilizando el canal, ya que cada nodo hace peticiones de las partes que le faltan. En el caso de usar una red Ad-hoc hay que tener en cuenta que la capacidad del medio es compartida por todos los nodos. Si todos los usuarios pretenden obtener el mismo contenido, en el canal habrán una gran cantidad de paquetes repetidos. Estas redes tienen un ratio de pérdida potencialmente alto para el consumo de contenidos en *streaming*, debido a que la variación en el retardo de recepción de las partes puede hacer que sean descartadas en el momento de la reproducción, y usan una arquitectura descentralizada compleja de gestionar en un entorno altamente dinámico.

I-A. Trabajo relacionado

Hay soluciones que proponen inundar la red de segmentos [5][6] lo cual solucionaría el problema de latencia y de pérdida, pero potenciaría la ineficiencia del uso del canal. Otras soluciones plantean utilizar algoritmos computacionalmente complejos [7] que harían que la latencia se viera incrementada.

La situación que prevemos es la descarga de un contenido por diferentes nodos pertenecientes a una red social, donde cada uno se descarga una parte proporcional. En otro instante los nodos se unen para formar una red Ad-hoc para intercambiarse las partes mediante un *streaming* P2P. Nuestra solución implementa un algoritmo de *gossip* en el cual los nodos/*peers* conocen periódicamente las partes del contenido que cada uno dispone. Para poder solventar la pérdida de partes proponemos un algoritmo de redundancia. Este algoritmo aprovecha la información y las estadísticas de la red social para generar una redundancia (mediante probabilidad de ocurrencia) que mejore el mecanismo de intercambio de P2P.

II. Solución propuesta

Este trabajo propone una solución para la distribución cooperativa de contenidos en entornos donde los usuarios forman parte de una misma red social, de manera que se puede aprovechar el conocimiento de su comportamiento y relaciones dentro del sistema.

II-A. *Escenario*

Los usuarios pertenecen a una red social, en la cual se despliega un servicio de distribución de contenidos. Mediante esta red social, los coordinadores de ésta y los propios usuarios podrán planificar eventos que requieran la presencia física de los usuarios y la visualización en común de determinados contenidos. Teniendo en cuenta que estos contenidos pueden requerir una alta tasa de descarga y que en el evento pueden participar múltiples usuarios, la descarga simultánea de dichos contenidos implicaría una gran capacidad del servidor si se utiliza una arquitectura centralizada.

Para paliar este problema, y gracias a la planificación de eventos (presenciales), se propone realizar una compartición de los costes de descarga entre los participantes haciendo un uso eficiente de los recursos disponibles, gracias al empleo de una arquitectura P2P (Figura 1).

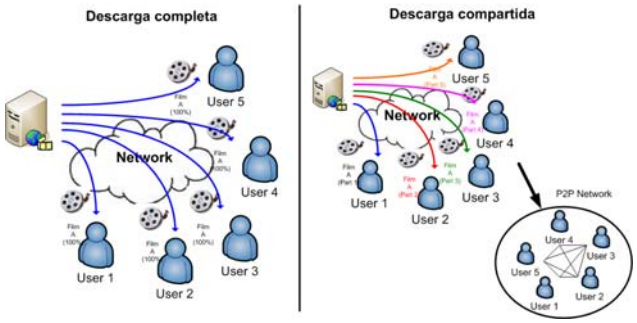


Figura 1. **Descarga completa**: cada usuario se descarga el fichero entero. **Descarga compartida**: cada usuario se descarga una parte equitativa del total del fichero.

El principal objetivo es disminuir al máximo la descarga concurrente, así como la cantidad de información descargada de un servidor central de contenidos con la finalidad de abaratar los costes de despliegue del servicio. Para ello, los participantes que asistan al evento programado recibirán una notificación de los contenidos que deberán descargar previamente a la realización del evento. De esta forma el servidor central enviará a cada participante una parte del contenido, distribuyendo así el coste de la descarga. En el momento de la realización del evento, los participantes compartirán las partes adquiridas con los demás usuarios sin consultar al servidor de contenidos, excepto en el caso de que la ausencia de uno o varios participantes implique que ciertas partes del contenido no estén disponibles en la red Ad-hoc.

La red social en la que se apoya el sistema de distribución permite que se tengan datos estadísticos acerca del comportamiento de los usuarios. Gracias a los datos obtenidos en eventos anteriores es posible estimar la probabilidad de asistencia de cada participante, de forma que los datos redundantes se pueden distribuir en función de este parámetro. Es decir, los participantes con mayor probabilidad de asistencia recibirán más información que los de menor probabilidad de asistencia. De esta forma se optimiza la utilización de datos redundantes.

II-B. *Mecanismo de distribución*

Para este trabajo se ha utilizado una implementación propia de un mecanismo de distribución P2P, inspirado en *CoolStreaming* [8]; una propuesta de red P2P que emplea un modelo *data-driven*. Este tipo de modelos son los más adecuados para realizar *streaming* de contenidos multimedia en entornos muy dinámicos y heterogéneos [9].

La distribución P2P está implementada mediante un protocolo de aplicación, que cuenta con dos planos independientes: control y datos.

El plano de control puede establecer, modificar y terminar relaciones de *membership*. Gracias a estas relaciones cada *peer* tiene una visión parcial de la red P2P. Cuando el plano de control arranca, automáticamente trata de descubrir nuevos miembros y de establecer relaciones de *membership* con ellos. Entre los participantes que descargan un mismo contenido se establece una relación de *partnership*. Una vez establecida esta relación, cada *peer* notifica de forma periódica las partes que posee de este contenido utilizando tablas llamadas *Buffer Map*. De esta forma cada participante puede crear una tabla con la información sobre las partes disponibles y los *peers* que las poseen.

El plano de datos se encarga de recibir y pedir a los demás *partners* las partes del contenido que desean visualizar. Para este escenario se plantean dos modos de funcionamiento. Al tratarse de eventos presenciales, los *peers* pueden formar una red Ad-hoc entre ellos. Hay que tener en cuenta que el canal es compartido por todos los usuarios, de forma que un mecanismo P2P basado en petición/respuesta *unicast* generaría una gran cantidad de tráfico en el medio. Por este motivo se plantea un modo principal de funcionamiento, en el que cada usuario reenvía a los demás las partes que él posee utilizando *broadcast*, optimizando el uso de los recursos de red. A este modo de funcionamiento se le incorpora el uso de técnicas de codificación de red (*Network Coding*) [10]. Este mecanismo permite mejorar el *throughput* en este tipo de redes, mediante la transmisión de una combinación lineal de los diferentes fragmentos que el *peer* desea enviar a la red. Es importante comentar que disponer de información redundante entre los *peers* de la red favorece el uso de técnicas de *Network Coding*, mejorando el número de posibles combinaciones lineales de paquetes a realizar. Esta correlación inicial provoca un aumento en la eficiencia de la transmisión, aumentando el número de paquetes codificados por transmisión, reduciendo la probabilidad efectiva de pérdida de un paquete, y finalmente reduciendo la carga que soportará la red Ad-hoc.

En paralelo al modo principal, se ejecuta un segundo modo de funcionamiento, el cual es una implementación de un sistema P2P tradicional. En este modo se utilizan las tablas creadas en el plano de control para descubrir el *peer* idóneo al que pedir las partes restantes. Su uso se limita al caso en que alguna de las partes no se hayan recibido a través de la distribución *broadcast* o no hayan podido ser recuperadas mediante *Network Coding*.

III. Cálculo de redundancia

En este trabajo se propone el cálculo de la redundancia necesaria a descargar por cada nodo en base al conocimiento recogido desde la aplicación de red social. El objetivo de introducir redundancia en la descarga parcial inicial del contenido en la red consiste en minimizar el impacto producido

por la posible falta de un determinado nodo en el momento de realizar la compartición de la información que cada nodo pone a disposición del resto en la red Ad-hoc.

Para ello, la aplicación que conforma la red social debe mantener un registro con información relativa a la fidelidad que un usuario tiene sobre sus eventos planificados. Es decir, la red social conoce los eventos a los que un usuario se suscribe (según su agenda o calendario compartido) y a los que realmente ha asistido. De este modo, se define la estimación de probabilidad de asistencia de un determinado nodo i, P_{n_i}, como el cociente de los eventos a los que ha asistido el usuario, e_{OK_i}, sobre todos los eventos a los que ha notificado asistir, e_{Total_i}.

$$P_{\mathrm{n}_i} = \frac{e_{\mathrm{OK}_i}}{e_{\mathrm{Total}_i}} \quad (1)$$

De este modo, se puede determinar la probabilidad de no asistencia de un nodo como el complementario del anterior.

$$\overline{P_{\mathrm{n}_i}} = 1 - P_{\mathrm{n}_i} = \frac{e_{\mathrm{Total}_i} - e_{\mathrm{OK}_i}}{e_{\mathrm{Total}_i}} = \frac{e_{\mathrm{KO}_i}}{e_{\mathrm{Total}_i}} \quad (2)$$

Una vez conocemos la estimación de la probabilidad de asistencia, se le asigna un peso específico a cada nodo, W_{n_i}, obtenido en función de ésta.

$$W_{\mathrm{n}_i} = \frac{P_{\mathrm{n}_i}}{\sum_{i=1}^{N} P_{\mathrm{n}_i}} \quad (3)$$

donde N representa el número de usuarios que han planificado asistir al evento.

El peso asignado a un determinado nodo representa el grado de confianza que se tiene sobre un determinado usuario. Por ello, un nodo con un peso mayor, obtendrá una parte mayor de redundancia que un nodo con poco peso. Por otro lado, con el objetivo de maximizar la presencia de información a compartir en el momento del evento, se define un umbral, $U_{\max}$, para limitar el número máximo de nodos que pueden fallar en el momento del evento (condición de contorno). $U_{\max}$ se obtiene fruto del siguiente algoritmo iterativo.

Algorithm 1 Obtención del umbral $U_{\max}$

```
int estimacionNodosFallan(Umax) {
  K=1;
  while (Pk > Umax)
  {
    Pk = Prob_fallo_K_nodos;
    K++;
  }
  return K;
}
```

Donde P_K (4) corresponde al cálculo de la probabilidad de que usuarios no estén presentes en el momento del evento, obtenido según la función de probabilidad binomial.

$$P_K = \binom{N}{K} \left(\overline{P_{\mathrm{n}}}\right)^K \left(P_{\mathrm{n}}\right)^{N-K} \quad (4)$$

donde $K \leq N$, y siendo, en este caso $P_{\mathrm{n}} = \frac{\sum_{i=1}^{N} P_{\mathrm{n}_i}}{N}$, y de manera complementaria, $\overline{P_{\mathrm{n}_i}} = 1 - P_{\mathrm{n}_i}$.

Finalmente, cada usuario se descargará la parte del contenido que le corresponde sobre la totalidad del fichero a compartir entre todos los usuarios planificados para el evento junto a una parte de redundancia calculada por la aplicación en base a la información obtenida de la red social (estadísticas de usuario). Por tanto, el porcentaje de información que cada usuario se descargará del fichero total a compartir, DL_i, se obtiene según la expresión siguiente.

$$DL_i\,(\%) = \left(\frac{1}{N} + \left(\frac{K}{N} \cdot W_{\mathrm{n}_i}\right)\right) \cdot 100 \quad (5)$$

IV. Caso de uso

El caso de uso propuesto se enmarca dentro de un entorno docente.

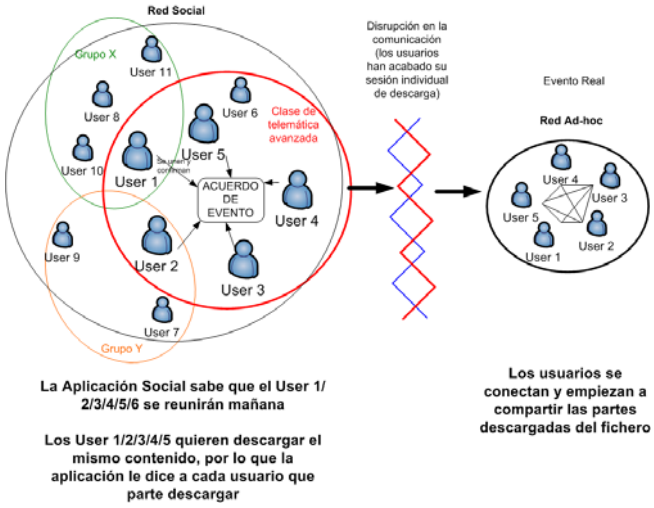


Figura 2. Escenario del caso de uso

El profesorado dispone de una herramienta de *e-learning* orientada a la comunidad educativa (red social educativa). Dentro de esta comunidad los diferentes usuarios (profesores, estudiantes, gestores de la aplicación) pueden interactuar de forma dinámica, usando foros, intercambiando mensajes, chats, creando eventos, compartiendo información, material didáctico, etc.

En este caso de uso, uno de los profesores de la comunidad desea compartir un nuevo contenido audiovisual con los estudiantes matriculados en su curso actual. Para ello el profesor planifica un evento dentro de la herramienta de *e-learning*, adjuntando un contenido multimedia audiovisual que será reproducido durante el desarrollo de la actividad. Este contenido se deposita en un repositorio privado donde posteriormente los usuarios autorizados deberán acceder para descargarlo (Figura 2).

Dado el elevado tamaño de los contenidos multimedia, no es recomendable que cada uno de los usuarios que desean consumirlo, lo descargue en su totalidad. Permitir dicho comportamiento provocaría un aumento en la infraestructura nece-

saria (número de servidores, ancho de banda) para mantener un servicio de calidad.

Este trabajo propone reducir al máximo esta problemática. Para ello, se propone la integración de diferentes técnicas, con el fin de ofrecer el contenido de forma segura y reduciendo al máximo la carga sobre la infraestructura. Por este motivo, se utilizan las estadísticas de asistencia a eventos obtenidas de la red social con el fin de introducir un cierto grado de redundancia de información en la red para poder realizar el intercambio de la información en el momento del evento.

Con esta información es posible optimizar el reparto de las partes del fichero sobre los usuarios, con dos objetivos finales. Por un lado, intentar garantizar la realización del evento. Por otro lado, favorecer la correlación de la información compartida entre los usuarios para mejorar los resultados de la utilización de los algoritmos de *Network Coding* (NC).

IV-A. Ejemplo

En la Figura 3 se reflejan diferentes porcentajes de descarga del fichero en función del número de usuarios que se prevé que no asistirán al evento programado, K. Para la obtención de estos resultados, se ha supuesto un escenario basado en el caso de uso contemplado en este trabajo. Se ha tomado una clase compuesta por 20 alumnos (N) y se ha definido un umbral $U_{\max}$ de asistencia del 20 % de los alumnos (en el sistema propuesto, este valor se obtendría del histórico de asistencia a clase).

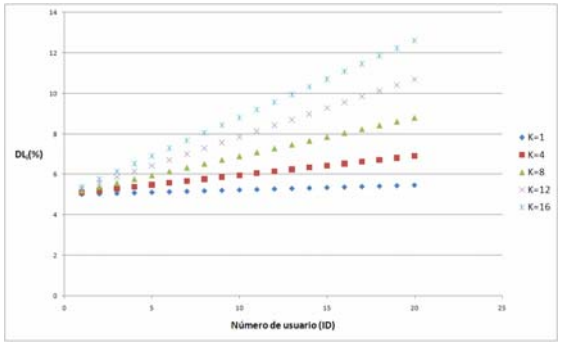


Figura 3. Porcentaje de descarga asignado a cada usuario

Esta gráfica muestra el porcentaje de carga asignado a cada usuario, donde la probabilidad de asistencia va del usuario uno que es el que menos tiene al 20 que es el que más. Se puede comprobar que a medida que el número *K* aumenta, el porcentaje a descargar DL_i por cada usuario también aumenta con el fin de aumentar la redundancia. También se comprueba que contra mayor sea la probabilidad de asistencia del usuario más porcentaje se le asigna.

V. Conclusiones

En este trabajo presentamos una solución novedosa para realizar de manera eficiente el *streaming* de vídeo en redes Ad-hoc P2P aprovechando la información obtenida de una red social. En base a la información obtenida de la red social, proponemos un modelo estadístico que permite hacer una estimación de los nodos que van a estar presentes en el momento del evento, y se explica cómo calcular la cantidad de información extra que cada nodo debe descargar con la finalidad de poder llevar a cabo satisfactoriamente la visualización de un determinado contenido multimedia cuando el evento tenga lugar. En este trabajo se propone el uso de *Network Coding* como técnica de distribución de contenido de media en redes Ad-hoc, el cual permite hacer un uso eficiente de los recursos de la red a la vez que mejoramos el *throughput* de la transmisión. Es importante destacar que la redundancia añadida favorece la correlación de la información compartida de forma que se mejoran los resultados del uso de *Network Coding*. Para ejemplificar mejor lo descrito, se plantea la aplicación del sistema en un entorno de *e-learning* en el que se planifica una clase en la que se visualizará un contenido, de manera que cada usuario se descarga una parte del contenido original y que, en el momento del evento, será compartido y reproducido por todos los alumnos presentes.

Agradecimientos

Este trabajo fue parcialmente financiado por la Fundació i2Cat y MCyT (Ministerio de Ciencia y Tecnología del Gobierno de España) bajo el proyecto TSI2007-66637-C02-01, el cual es parcialmente financiado por FEDER.

Referencias

[1] Mitra, S.; Bagchi, A.; Bandyopadhyay, A.K. *Design of a Data Model for Social network Applications*. Journal of Database Management, Vol. 18, Issue 4.

[2] Haddon, L.; Dong Kim, Shin. *Mobile Phones and Web-based Social Networking – Emerging Practices in Korea with Cyworld*. The Journal of The Communications Network, Vol. 6, Parte 1, January–March 2007.

[3] Chang, Y.; Liu, H.; Chou, L.; Chen Y.; Shin, H. *A general architecture of mobile social network services*. 2007 International Conference on Convergence Information Technology - ICCIT '07, 21-23 Nov. 2007, Gyeongju, South Korea.

[4] Zhao, Y.; Hou, X.; Yang, M.; Dai, Y. *Measurement Study and Application of Social Network in the Maze P2P File-Sharing System*. ACM International Conference Proceeding Series, Vol. 152, Artículo Num. 57, 2006.

[5] Jin, S.: *Replication of Partitioned Media Streams in Wireless Ad Hoc Networks*. Proc. of the 12th ACM Int. Conf. on Multimedia. ACM press, New York (2004) 396 - 399.

[6] Shahram, G., Bhaskar, K. and Song, S.: *Placement of Continuous Media in Wireless Peer-to- Peer Networks*. IEEE Trans Multimedia, Vol. 6, No. 2. IEEE Computer Society, New Jersey (2004) 335-342.

[7] Zhuo, D.; Du, X.; Yang, Z. *Hybrid Search Algorithms for P2P Media StreamingDistribution in Ad Hoc Networks*. Computational Science-ICCS 2007. 7th International Conference. Proceedings, Part IV, 27-30 May 2007 , Beijing, China.

[8] Xinyan Zhang; Jiangchuan Liu; Bo Li; Yum, Y.-S.P., "CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming", INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE , vol.3, no., pp. 2102-2111 vol. 3, 13-17 March 2005.

[9] Thomas Silverston and Oliver Fourmaux, "Source vs Data-driven Approach for Live P2P Streaming" Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06), 2006.

[10] Gkantsidis C.; Rodriguez, P. *Network coding for large scale content distribution*. Proc. IEEE INFOCOM, Miami, FL, Mar. 2005, pp. 2235–2245.

Nuevo criterio para la estimación de información de estado de certificados en MANET

J. Muñoz, J. Parra-Arnau, C. Gañán, M. Jáimez
Departamento de Ingeniería Telemática. Universidad Politécnica de Cataluña (UPC)
C/ Jordi Girona 1-3. Campus Nord, UPC, 08034 - Barcelona.
Email: {jose.munoz, javier.parra, carlos.ganan, marc.jaimez}@entel.upc.edu

***Resumen*—In general, certificate status validation is a critical operation but it is particularly complex in Mobile Ad-hoc Networks (MANETs). MANET users require solutions to manage both the lack of fixed infrastructure inside the network and the possible absence of connectivity to trusted authorities when the certification validation has to be performed. However, certificate validation implies checking the validity of certificates in realtime, that is, when a particular certificate is going to be used. In such MANET environments, a node might be disconnected from the status data source when queries are required. Proposals in the literature suggest the use of caching mechanisms so that the node itself or a neighbor node has some status checking material. In this paper, we analyze how to deploy a certificate status checking PKI service for MANET. Besides, it is proposed a novel criteria that is much more appropriate and absolute than time. This criteria takes into account the revocation process and is based on risk to evaluate cached status data.**

I. Introducción

Las redes MANETs son redes cooperativas que permiten a los nodos inalámbricos establecer comunicaciones de una forma espontánea. Como se afirma en [1], se prevé que estas redes tengan topologías multisalto dinámicas, a menudo rápidamente cambiantes y aleatorias, y probablemente compuestas por enlaces inalámbricos limitados en ancho de banda. Las redes MANET pueden operar de manera autónoma o bien utilizando gateways a redes fijas. En este último caso, la red MANET recibe el nombre de "híbrida". Se espera que las redes MANET se desplieguen como una extensión de las redes de infraestructura tradicionales. Cabe mencionar que el comportamiento híbrido puede ser temporal debido a una situación en la que la red ad-hoc puede estar operando unas veces de forma autónoma y otras conectada a Internet. El escenario considerado en este artículo se basa en las redes MANET híbridas, que se prevé que se impongan en un futuro.

Por otro lado, la confianza y la seguridad son requerimientos básicos para soportar aplicaciones de negocios en este escenario. El esquema de clave pública es el mecanismo subyacente preferido para proporcionar servicios de seguridad. En un esquema de clave pública cada participante tiene dos claves: una clave pública (i.e. conocida por todos) y una clave privada (i.e. secreta). El anuncio de la clave pública se realiza mediante un documento firmado conocido como Public Key Certificate (PKC) o simplemente "certificado", que liga al participante con su clave pública. La entidad que firma el certificado recibe el nombre de "emisor de certificado" o "Certificate Authority" (CA). En la literatura existen varias formas de gestionar la seguridad y la confianza en las redes MANET que utilizan criptografía de clave pública, pudiéndose clasificar en base al grado de descentralización de los mecanismos desplegados para la emisión, publicación y revocación de los certificados. En este artículo nos hemos centrado en los esquemas centralizados, que consideramos apropiados para redes MANET híbridas en las que se requiere interoperatibilidad con las Public Key Infrastructures (PKIs) actualmente desplegadas.

Por otra parte, un certificado podría estar revocado (invalidado) antes de que expirase. Entre otras razones, un certificado puede estar revocado porque ha perdido la clave pública asociada o ésta se encuentra comprometida. Las políticas de revocación determinan cómo el estado de los certificados se distribuye a los usuarios finales. Existen un par de mecanismos estándares para la gestión de revocaciones explícitas. El mecanismo más simple consiste en emitir periódicamente una lista de certificados revocados o CRL (Certificate Revocation List) ([2], [3]). Una CRL es una lista "negra" de los identificadores de los certificados que se han revocado. El otro esquema es el Online Certificate Status Protocol (OCSP) [4], en el que la distribución de los datos de estado se lleva a cabo mediante la consulta a unas autoridades de confianza intermedias llamadas *responders*.

El principal inconveniente que se presenta al considerar esquemas centralizados es la dificultad que entraña la adaptación para redes MANET híbridas de soluciones que fueron originalmente diseñadas para redes cableadas y bien conectadas. En este nuevo entorno, se espera que los usuarios móviles se desplacen por distintas redes. Cuando un usuario se encuentre en una red con conexión a la PKI, éste podrá disponer de todos sus servicios, tales como conseguir un certificado, lanzar una consulta de estado, etc. Sin embargo, los usuarios podrían estar desconectados de la PKI cuando necesitaran un servicio PKI en tiempo real. En este sentido, la comprobación del estado del certificado es un servicio crítico, ya que las aplicaciones deben decidir, en el momento en el que se va a utilizar, si un certificado es válido o si no se puede realizar una determinada acción. Para tomar una decisión, el usuario sólo dispone de información de estado del certificado en el momento en el que fue emitida.

Las diferentes propuestas en la literatura ([5], [6] y [7]) sugieren el uso de mecanismos de *caching* que permitan gestionar desconexiones arbitrarias entre los usuarios y las fuentes del servicio de datos de estado. Las desconexiones se palian por medio del almacenamiento de copias de los datos de estado (listas de certificados revocados o respuestas online) en los nodos de la red ad-hoc.

De acuerdo con [8], existen dos mecanismos para que los usuarios comprueben la frescura de la información de estado del certificado. El primero utiliza *nonces*, que son adecuados para un escenario donde pueden ocurrir desconexiones, y el

segundo está basado en el tiempo transcurrido desde que se emitió la información de estado del certificado. En este artículo proponemos y formulamos un nuevo criterio basado en el conocimiento del proceso de revocación global dentro de la red MANET. Este método permite evaluar los datos de estado cacheados mediante el cálculo de la probabilidad de considerar un certificado como válido cuando el estado real conocido por la PKI, en un instante dado, es el de revocado. Tal y como detallaremos más adelante, este criterio es mucho más apropiado y absoluto que el basado en el tiempo.

II. Evaluación de los Datos

Cuando la CA emite los datos de estado de los certificados incluye dos sellos temporales:

- *thisUpdate.* Instante en el que los datos de estado han sido emitidos.
- *nextUpdate.* Instante en el que se espera que se emitan datos de estado actualizados.

Definamos T_s como el intervalo de emisión de los datos de estado:

$$T_s = nextUpdate - thisUpdate \quad (1)$$

Como los datos de estado están ligados a estos dos sellos temporales, los usuarios pueden tener una idea de la frescura del estado de un certificado inspeccionando *thisUpdate*. Así, los usuarios pueden tomar la decisión de operar o no con este certificado. A nuestro saber, éste es el único criterio propuesto en la literatura para ayudar al usuario a tomar una decisión. En nuestra opinión, la evaluación en base al tiempo de los datos cacheados es un criterio que aporta poca información. En esta sección proponemos otro parámetro para esta evaluación.

En primer lugar, ilustremos por qué el tiempo es un parámetro pobre para nuestros propósitos. Por ejemplo, consideremos una respuesta de estado emitida hace un par de horas. Podemos preguntarnos: *¿es fresca o no?*. Obviamente, la respuesta es *"depende"*. No se pueden considerar dos horas como un largo periodo de tiempo si hay un par de certificados revocados al mes, pero se puede considerar este periodo bastante largo si hay dos nuevos certificados revocados por hora. Asimismo, un escenario con millones de certificados emitidos no expirados no es el mismo que otro con cientos de certificados. En el primero, un par de nuevos certificados revocados no es relevante, mientras que en el último, este mismo número de certificados sí que es importante. Como conclusión, necesitamos un parámetro que considere todos estos aspectos. Para nuestro propósito, definimos una función de *riesgo* que ayuda al usuario a decidir si puede confiar en un certificado o no. Formalmente definimos la función *riesgo* ($r(t)$) como la *probabilidad de considerar un certificado como válido cuando su estado real conocido por la PKI es revocado en el instante* t.

Para encontrar una expresión analítica de la función *riesgo* primero necesitamos analizar el proceso de emisión de certificados. Los certificados son emitidos con un periodo de validez T_c. Obviamente, $T_c >> T_s$; por ejemplo, T_c puede ser un año, mientras que el periodo de emisión de los datos de estado puede ser de una hora. El número de *certificados no expirados* ($N(t)$), incluyendo tanto a los revocados como a los no revocados, es un proceso estocástico cuyo valor medio en el instante *t* depende de los procesos de emisión y expiración de certificados. Se asume que el tiempo transcurrido desde la emisión hasta la expiración (T_c) es un valor constante para todos los certificados. Por tanto, el proceso de expiración es el mismo que el proceso de emisión transcurridas T_c unidades de tiempo. Se asume un proceso de *Poisson* para la emisión de certificados puesto que:

- Cada emisión es independiente de la anterior (*sin memoria*). El hecho de que se produzca una emisión en un instante determinado no dice nada sobre la probabilidad de una emisión en un instante anterior o posterior. No se puede predecir la próxima emisión a partir de información actual o anterior.
- En nuestro escenario de trabajo se considera que la población de usuarios que solicitan un certificado es relativamente grande. Así, la tasa media de peticiones es independiente de la ventana temporal. Por consiguiente, esta tasa es constante λ_c.
- La probabilidad de que un usuario solicite un certificado es proporcional al tiempo, i.e. $\lambda_c \Delta t + O(\Delta t)$.

Al satisfacer estas tres propiedades, el proceso considerado es conocido como proceso de *Poisson*. Este proceso queda definido por su tasa de emisión de certificados λ_c, que se corresponde con la tasa de expiración de certificados. De esta manera, el valor medio de *certificados no expirados* en régimen permanente es el número medio de certificados emitidos antes de que empiece el proceso de expiración.

$$E[N(t)] = N = \lambda_C T_C, \quad t > T_C \quad (2)$$

Por otro lado, existe un grupo de *certificados revocados no expirados*, es decir, certificados que tienen un periodo de validez correcto pero que han sido revocados antes de la fecha de expiración y, por tanto, están incluidos en la lista negra. El subconjunto de *certificados revocados no expirados* están incluidos en el conjunto de *certificados no expirados* y el cardinal de ese conjunto, $R(t)$, es un proceso estocástico que típicamente se modela [9] como una fracción o porcentaje ($p(t)$) de los certificados no expirados:

$$R(t) = p(t)N(t) \quad with\ p(t) \leq 1 \quad (3)$$

Asumiendo que ambos procesos son independientes y utilizando valores medios:

$$E[R(t)] = E[p(t)]E[N(t)] \quad (4)$$

$$R = pN \quad (5)$$

Modelamos el porcentaje esperado de certificados revocados como directamente proporcional al tiempo de certificación T_c:

$$p = p' T_c \quad (6)$$

Esto significa que periodos de certificación más grandes conllevan un mayor porcentaje de certificados revocados. Por otro lado, periodos de certificación más pequeños implican una probabilidad menor de que un certificado sea revocado durante su periodo de vida y, por tanto, un menor porcentaje de certificados revocados. De esta forma, el valor medio

de *certificados revocados no expirados* puede ser expresado como:

$$R = p'\lambda_c T_c^2 \tag{7}$$

Llegados a este punto, hemos modelado el proceso de emisión y de revocación del sistema global. Sin embargo, nuestro objetivo es modelar el *riesgo* desde el punto de vista del usuario, o sea, queremos encontrar la probabilidad de considerar un certificado como válido cuando el estado real conocido por la PKI es revocado.

Asumamos, sin pérdida de generalidad, que en el instante $t_0 = thisUpdate$ un usuario consigue la lista negra actual de certificados revocados de la PKI. Utilizando esta lista, el usuario puede dividir el conjunto *certificados no expirados* en *certificados revocados* y *certificados no revocados*.

A continuación, definimos el subconjunto de *certificados operativos* como el conjunto de *certificados no expirados* para el cual el último estado conocido por el usuario era *no revocado*. Conviene percatarse de que la PKI puede saber que un certificado considerado como operativo por un usuario puede estar, en realidad, revocado. Sin embargo, dada la naturaleza de la red MANET, podría no ser capaz de comunicar esta situación al usuario.

Ahora asumamos que el usuario ya no es capaz de conectarse a la infraestructura. A medida que el tiempo avanza, el conjunto de *certificados operativos* incluirá certificados revocados y el usuario necesitará tomar decisiones sobre si usar un certificado operativo asumiendo un cierto *riesgo*. La función *riesgo* $r(t)$ puede ser evaluada como la ratio entre el número de *certificados operativos revocados desconocidos* ($R'(t)$) y el número de *certificados operativos* ($N'(t)$), tal y como se muestra en la ecuación:

$$r(t) = \frac{E[R'(t)]}{E[N'(t)]} \tag{8}$$

$N'(t)$ (*número de certificados operativos*) puede ser definido como el número de certificados que no fueron incluidos en la última lista negra obtenida por el usuario (estaban no revocados antes de t_0) y que no han expirado en t. Incluido en el conjunto de *certificados operativos* existe un subconjunto de *certificados operativos revocados no conocidos*. El cardinal de este subconjunto $R'(t)$ es el número de *certificados operativos* que están revocados en el instante t, es decir, están revocados pero este hecho es desconocido para el usuario.

En el instante $t_0 = thisUpdate$, el conjunto de *certificados operativos* es el mismo que el del conjunto de *certificados no revocados* y, puesto que el usuario tiene la misma información que la PKI, no hay *riesgo* ($r(t_0) = 0$). Además,

$$E[N'(t_0)] = (1-p)N \tag{9}$$

$$E[R'(t_0)] = 0 \tag{10}$$

En el instante $t_0 + T_C$ todos los certificados incluidos en la lista negra habrán expirado. Esto significa que todos los *certificados no expirados* serán *operativos*, y que ningún certificado revocado será desconocido para el usuario. El *riesgo* en este momento puede ser expresado como:

$$r(t_0 + T_C) = \frac{E[R'(t_0 + T_C)]}{E[N'(t_0 + T_C)]} = \frac{E[R(t_0)]}{E[N(t_0)]} = p \tag{11}$$

Para evaluar la función *riesgo* entre t_0 y $t_0 + T_C$ debemos observar los procesos $N'(t)$ y $R'(t)$ en este intervalo. Después de t_0, la variación del número de *certificados operativos* ($N'(t)$) depende de estos factores:

- Incrementa debido a nuevas emisiones.
- Decrementa debido a la expiración de certificados operativos que fueron emitidos antes del instante t_0 (los certificados emitidos más tarde no expiran en el intervalo considerado).

La tasa de emisión es λ_c, que es la misma que la tasa de expiración. Sin embargo, cabe destacar que no todas las expiraciones conciernen a *certificados operativos*. Una fracción p de las expiraciones corresponde a *certificados revocados no expirados*, y la otra fracción $1-p$ corresponde a *certificados operativos*. Entonces, la tasa de expiración de *certificados operativos* es $(1-p)\lambda_c$ (véase figura 1).

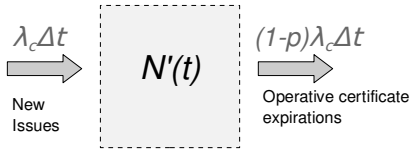


Figura 1. Evolución de certificados operativos

Considerando la evolución del conjunto de *certificados operativos* podemos evaluar su cardinal medio:

$$E[N'(t)] = E[N'(t_0)] + \lambda_C(t-t_0) - (1-p)\lambda_C(t-t_0) \tag{12}$$

Usando (9) se obtiene:

$$E[N'(t)] = (1-p)N + p\lambda_C(t-t_0) \tag{13}$$

Finalmente, se necesita una expresión para el conjunto de *certificados operativos revocados*. Este conjunto es la intersección del conjunto de *certificados operativos* y el conjunto de *certificados revocados*, como se muestra en la Figura 2.

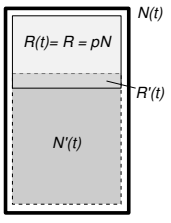


Figura 2. Conjuntos de certificados

Así, podemos expresar la cardinalidad de estos conjuntos usando la siguiente expresión:

$$N(t) = R(t) + N'(t) - R'(t) \tag{14}$$

Por lo tanto,

$$R'(t) = R(t) + N'(t) - N(t) \quad (15)$$

Obtenemos el valor medio del número de certificados operativos revocados usando (2), (5), (13) y (15):

$$E[R'(t)] = p\lambda_C(t - t_0) \quad (16)$$

Para obtener la función analítica del *riesgo* se usan las expresiones (13), (16) y su propia definición:

$$r(t) = \frac{p(t - t_0)}{(1-p)T_c + p(t - t_0)} \quad (17)$$

La expresión previa es válida para instantes de tiempo $t \ \epsilon \ t_0 \leq t \leq t_0 + T_c$ y cumple con los resultados esperados de las expresiones (10) y (11). Cabe destacar, que la función del *riesgo* permite a un usuario calcular la probabilidad de considerar un certificado no expirado como no revocado cuando el estado real conocido por la PKI es de revocado.

Por otra parte, es remarcable que, a diferencia del tiempo, que es un parámetro relativo, la función *riesgo* proporciona al usuario un parámetro absoluto que le ayuda a tomar la decisión de confiar o no en un certificado concreto. Esta decisión se debe tomar cuando el usuario está desconectado de la infraestructura y por lo tanto está teniendo en cuenta información de estado cacheada (i.e obsoleta).

Finalmente, la función *riesgo* debe usarse de la siguiente manera:

- En primer lugar, la CA firma la información de estado con los dos sellos temporales estándares (*thisUpdate* y *nextUpdate*) pero también añade el parámetro actual p. La CA puede calcular este parámetro ya que conoce el número actual de certificados emitidos que no han expirado y el número actual de certificados revocados que tampoco han expirado.
- Cuando un usuario tiene que evaluar información de estado, éste conoce T_c ya que es el periodo de certificación que está adjunto en el certificado.
- Así, el usuario obtiene p de la información de estado.
- Después, el usuario puede calcular el *riesgo* en el instante actual t reemplazando t_0 con *thisUpdate* en la función *riesgo*.
- Finalmente, el usuario puede tomar una decisión sobre un certificado en concreto con el valor de *riesgo* que ha calculado.

III. Conclusiones

Las arquitecturas de certificación descentralizadas para MANET, en general, proporcionan mecanismos para la validación de certificados dentro de la MANET. Sin embargo, la validación local de certificados y la interoperatibilidad con PKIs ya desplegadas puede restringir su usabilidad en un escenario MANET híbrido. Si se usa una infraestructura de certificación centralizada, entonces la validación de certificados se convierte en un problema a tener en cuenta. Esto se debe a que los usuarios necesitan asegurarse en el momento de uso que los certificados en los que ellos confían no han sido revocados. Sin embargo, en este mismo momento los servidores de confianza de la PKI pueden no estar accesibles. Además, los mecanismos de comprobación de estado estándares para redes fijas no son aplicables de forma directa, ya que fueron diseñados para usuarios siempre conectados.

En este sentido, los esquemas de *caching* permiten gestionar desconexiones arbitrarias entre los usuarios y las fuentes de servicios de datos de estado. Las desconexiones se palian mediante el almacenamiento de las copias de los datos de estado (listas de certificados revocados y respuestas online) en los nodos de la red ad-hoc. Estas copias se obtienen cuando la conexión a la infraestructura está disponible. En este artículo, hemos estudiado y analizado todos estos problemas para adaptar los mecanismos estándares de comprobación de datos de estado de PKI a MANET.

A pesar de que los esquemas de *caching* permiten a los usuarios obtener datos de estado durante las desconexiones, la información cacheada es probable que esté anticuada. Cuando se usa información de estado cacheada un nodo puede que opere con un certificado revocado considerándolo como válido. En este artículo, hemos presentado un esquema nuevo que proporciona a los usuarios que pertenecen a la MANET un criterio absoluto para determinar si usar o no un certificado en concreto cuando no se dispone de información de estado actualizada. Teniendo en cuenta información acerca del proceso de revocación, los usuarios pueden calcular una función *riesgo* para estimar si se ha revocado un certificado mientras no había conexión a un servidor para comprobar su estado. Finalmente, también cabe mencionar que este nuevo criterio puede aplicarse a otras redes que no sean MANETs, si estas redes se basan en esquemas de revocación explícita.

Referencias

[1] S. Corson and J. Macker. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. RFC 2501, Internet Engineering Task Force, January 1999.

[2] R. Housley, W. Ford, W. Polk, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and CRL Profile. RFC 2459, Internet Engineering Task Force, January 1999.

[3] S. Tuecke, V. Welch, D. Engert, L. Pearlman, and M. Thompson. Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile. RFC 3820, Internet Engineering Task Force, June 2004.

[4] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC 2560, Internet Engineering Task Force, June 1999.

[5] L. Yin and G. Cao. Supporting cooperative caching in ad hoc networks. *IEEE Transactions on Mobile Computing*, 2006.

[6] H. W. Go, P. Y. Chan, Y. Dong, A. F. Sui, S. M. Yiu, L. C. K. Hui, and V. O. K. Li. Performance evaluation on crl distribution using flooding in mobile ad hoc networks (manets). In *ACM Southeast Regional Conference archive. Proceedings of the 43rd annual southeast regional conference*, Kennesaw, Georgia, 2005.

[7] G. F. Marias, K. Papapanagiotou, V. Tsetsos, O. Sekkas, and P. Georgiadis. Integrating a trust framework with a distributed certificate validation scheme for manets. *Wireless Communications and Networking*, 2006.

[8] A. Deacon and R. Hurst. The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments. RFC 5019, Internet Engineering Task Force, September 2007.

[9] A. Arnes. Public key certificate revocation schemes, 2000. Queen's University. Ontario, Canada. Master Thesis.

Análisis de consumo energético y tiempo de proceso en cifrado AES en Redes Inalámbricas de Sensores

Lourdes López Santidrián, Vicente Hernández Díaz, Alberto Maján Cortijo, José Fernán Martínez Ortega, Ana Belén García Hernando y Antonio Dasilva Fariña.
Departamento de Ingeniería y Arquitecturas Telemáticas (DIATEL)
EUIT de Telecomunicación. Universidad Politécnica de Madrid
Campus Sur UPM. Ctra. de Valencia, km. 7 28031 Madrid.
E-mail: { llopez, vhernandez, amajan, jfmartin, abgarcia, adasilva}@diatel.upm.es

***Resumen*- Las redes inalámbricas de sensores, formadas por una serie de pequeños dispositivos llamados nodos o motas, tienen muchas limitaciones en sus capacidades de procesamiento y de memoria, por lo que resulta compleja la utilización de algoritmos de cifrado con esta tecnología. Por esta razón, este artículo se ha centrado en la evaluación del uso del estándar de criptografía de clave simétrica AES, a través del diseño e implementación de un escenario que permitiera analizar su comportamiento sobre una aplicación real: control de seguridad de un perímetro que alerta sobre la presencia de intrusos a través de las variaciones en la luz y en el nivel de vibración. Sobre este sistema, se han realizado medidas del consumo de energía y de los recursos de procesamiento necesarios, con el fin de obtener una estimación real del consumo que implica la aplicación de un algoritmo de cifrado simétrico en un entorno real de aplicación.**

***Palabras Clave*-criptografía de clave simétrica, AES, evaluación de tiempo de procesamiento, evaluación de energía, redes inalámbricas de sensores.**

I. INTRODUCCIÓN

Las Redes Inalámbricas de Sensores (WSN *Wireless Sensor Networks*) ofrecen un marco idóneo para desarrollar aplicaciones y proporcionar servicios que den respuesta a mediciones realizadas por diversos sensores, permitiendo una toma de decisiones autónoma sin un despliegue de red específico y sin una jerarquía de nodos concreta.

Los ámbitos de aplicación son muy variados y extensos, y según se va avanzando en el desarrollo de aplicaciones y servicios sobre WSN, se va haciendo más necesaria la implantación de servicios de seguridad en este tipo de redes [1], entre los que cabe destacar, los servicios de confidencialidad, autenticación e integridad.

Las restricciones de energía, de capacidad de cómputo y de capacidad limitada de comunicaciones que particularizan a estas redes hace necesario plantearse si los algoritmos tradicionales en los que se apoyan los mecanismos de seguridad son óptimos o no en estos entornos.

El objetivo principal de este artículo es analizar si el actual estándar de cifrado de clave simétrica AES (*Advanced Encryption Standard*) [2], resulta adecuado para proporcionar servicios de confidencialidad, autenticación e integridad en WSN.

Mediante la realización de este estudio se pretende constatar el impacto que supone la incorporación de estos servicios de seguridad sobre el consumo de batería de las motas o el tiempo de procesamiento que requieren, en un entorno de funcionamiento real.

El artículo se divide en cinco apartados. En este primer apartado se presenta la motivación que ha conducido a la realización del estudio que se presenta en este artículo.

En el segundo se realiza una introducción para acercar al lector a las redes de sensores inalámbricos y la problemática que presentan con respecto a sus limitados recursos y el uso adecuado que se ha de hacer con ellos.

La tercera parte de este artículo está dedicada a la descripción de la aplicación sobre la que se ha realizado un estudio práctico, en el que se realizaron medidas sobre una WSN para determinar el impacto que supone la inclusión de servicios de seguridad basado en AES sobre estas redes.

En la cuarta parte del artículo se presentan los resultados obtenidos en la aplicación de cifrado AES, los cuales se cuantifican en términos de consumo de energía y de tiempo de procesamiento.

Por último, se presenta el apartado de conclusiones obtenidas tras la realización de este estudio.

II. REQUISITOS FUNCIONALES Y NO FUNCIONALES DE LAS REDES INALÁMBRICAS DE SENSORES

La disposición típica de una WSN consiste en la dispersión de sus nodos o motas en una zona sobre la cual se desean realizar determinadas mediciones. Las motas se coordinan entre sí para generar información de alta precisión sobre el entorno físico. Cada mota basa las decisiones que tiene que tomar en tres parámetros básicos: la misión que se le ha encomendado, la información que tiene actualmente y sus recursos actuales (tanto a nivel de comunicaciones, como computacionales o energéticos). En función de la topología de la WSN [3], cada mota desplegada tendrá la capacidad de recolectar información y enviarla a otras motas, para el caso de una topología descentralizada, o bien, de enviar la información a un único nodo central, denominado sumidero, para el caso de una distribución en forma de estrella.

En cualquiera de las posibles aplicaciones para las que se diseñan las WSN, aparecen dos problemas principales y que serán críticos: el consumo de energía y la cantidad de recursos hardware de los que dispone la mota [4]. Los nodos

están muy limitados energéticamente, puesto que tienen que depender de baterías cuya duración es finita y en la mayoría de los casos no se pueden recargar o sustituir. Debido a esto, el diseño de algoritmos eficientes energética y computacionalmente ha llegado a ser un factor clave para dotar a las WSN de un tiempo de vida elevado. Con respecto a los recursos hardware, no debemos olvidar que las motas son dispositivos sencillos, con una reducida cantidad de memoria y una capacidad de procesamiento reducida. Por ejemplo, una mota típica tiene 16-bit, 8 MHz RISC CPU con aproximadamente 10K de RAM, 48K de ROM y 1024K de almacenamiento flash.

Las dos acciones que requieren un mayor consumo de energía por parte de las motas son el procesado de las señales y las comunicaciones. Debido a que en los nodos la batería es un factor limitante, mantener activas estas funciones, reduciría su tiempo de vida drásticamente. La organización y gestión óptimas de la red de sensores será crucial para garantizar un nivel aceptable de rendimiento y para mantener la energía de la mota el tiempo suficiente para llegar a finalizar la misión requerida. Una correcta organización de la red deberá activar sólo un subconjunto de motas en caso que se produzca un determinado evento en sus inmediaciones, manteniendo el resto de la red en estado de hibernación ya que su activación no sería necesaria. Para conseguir el nivel de rendimiento deseado, la gestión de la energía será un tema vital para alargar la vida de la red.

Las limitaciones de energía combinadas con el despliegue típico de un gran número de sensores, han hecho necesaria la implementación de métodos de ahorro de energía en todas las capas de la pila de protocolos [5]. Una de las mayores limitaciones con respecto a la vida de la batería se puede encontrar en las conexiones vía radio, puesto que son una de las acciones que más consumo requieren [6]. Este consumo es especialmente significativo cuando la radio del receptor se encuentra activa todo el tiempo. La energía consumida durante las transmisiones de radio es directamente proporcional a la distancia de alcance de la misma.

Otro apartado que hasta hace pocos años no se había podido abordar es el de la utilización de algoritmos criptográficos en las motas. Gracias al incremento de las capacidades de procesamiento y de memoria de las motas, se ha abierto el camino para la utilización de algoritmos criptográficos aplicados a la confidencialidad, integridad y autenticación, especialmente los de clave simétrica. De acuerdo con [7], algoritmos de cifrado simétrico como RC5, RC6, TEA o SkipJack resultan adecuados para ser utilizados en este tipo de redes debido a sus pocos requerimientos de capacidad de procesamiento y su consumo razonable de energía. Sin embargo, en este mismo estudio se indica que el antiguo estándar de cifrado DES, no resulta adecuado para este tipo de redes. El estudio comparativo entre AES 128 bits y SHA-1 [8] demuestra que ciertas implementaciones del algoritmo AES pueden considerarse efectivas para proporcionar servicios de seguridad basados en cifrado.

III. CASO DE USO: APLICACIÓN DE CONTROL DE ACCESO PERIMETRAL

Para la realización de este estudio se han aplicado servicios de seguridad basados en criptografía de clave simétrica sobre una aplicación desarrollada para realizar un control de acceso perimetral en una instalación de alta seguridad. Como requisito de la aplicación se plantea que ninguna persona pueda entrar en el área restringida sin que el sistema informe de que se ha producido esa vulneración de la seguridad.

Debido a que se trata de una instalación de alta seguridad, el empleo de un algoritmo de cifrado es necesario puesto que se quiere proteger la confidencialidad de las comunicaciones, se quiere garantizar que no se producen modificaciones de las mismas y es necesario autenticar a los nodos que están enviando la información. Suponiendo que personas ajenas la instalación quieran entrar en una zona restringida, es de esperar que busquen un medio para poder lograrlo, es decir un fallo en la seguridad.

Para realizar el control de presencia se trabaja con dos magnitudes físicas: por un lado, la cantidad de luz que recibe la mota, de modo que si se produce una variación de más de un 10% en la cantidad de luz que se recibe el nodo en ese momento, se disparara una alarma. Cabe reseñar que esta variación puede ser tanto positiva como negativa, es decir, si un objeto pasa por delante de la mota reduciría la cantidad de luz que recibe, pero si por el contrario, el nodo es alumbrado con algún tipo de luz como una linterna, la cantidad de luz aumenta. Por otro lado, se trabaja con el control de las vibraciones. Si la mota se acopla a una alambrada o se sitúa en el mismo suelo, cuando alguien o algo toca la verja o pasa cerca del nodo, este detectará las vibraciones que se producen y lo notificará de inmediato activando una alarma.

Para el desarrollo de la aplicación se eligió por simplicidad, una topología en estrella, en la que un conjunto de motas se comunican con un único sumidero conectado a un ordenador que es el encargado de procesar la información recibida desde la red de sensores. Si bien este es un tipo de topología muy sencilla, cumple adecuadamente con los requisitos que se fijaron para la realización de este estudio y además era la que más se ajustaba al escenario propuesto.

A continuación se detallará el software específico de cada elemento de la red, teniendo en cuenta las tareas que se quiere que realice cada dispositivo y cómo debe llevarlas a cabo, según las premisas del escenario de uso.

Debido a que en principio se desconoce el tiempo de vida de la aplicación, interesa que sea lo máximo posible, por lo tanto, las motas que actúan como sensores deben permanecer inactivas constantemente, para de ese modo, aumentar al máximo su vida útil, y sólo activarse en caso de producirse alguna perturbación. Se entiende por perturbación la activación de algunos de los sensores que interesan en esta aplicación, como son el sensor de luz y el acelerómetro. En ese momento, se comprueba si la variación producida en las lecturas supera un determinado margen previamente establecido. En caso de ser así se prepara un paquete, se cifra, se le añade un código MAC y se envía al sumidero.

El sumidero se ha de encargar de autenticar a las motas, de recibir la información que le envían, descifrarla y de comprobar que no ha sido alterada durante su envío. Una vez procesada la información se envía al ordenador de procesamiento para gestionar la información recibida. El ordenador de procesamiento es el encargado de procesar la información y de presentarla en pantalla, para ello se ha implementado una interfaz gráfica que facilita la comprensión de los datos recibidos.

Para la realización del estudio de eficiencia del algoritmo AES se ha decidido usar AES de 256 bits, dado que las claves son muy robustas y las operaciones que realiza son muy básicas y se permite la optimización de recursos del sistema. Las motas empleadas son el modelo Imote2 de Crossbow. Este modelo es uno de los últimos que ha aparecido en el mercado y dispone ya de una importante capacidad de cómputo y de almacenamiento. Dispone de un procesador Intel PXA271 XScale, con una frecuencia de trabajo seleccionable entre 13 y 416 Mhz (las pruebas se realizaron con esta última), 256 kB SRAM, 32MB SDRAM y 32 MB de memoria flash. Por otro lado, dicho modelo ofrece la posibilidad de desarrollar aplicaciones en las motas para diversas plataformas software: Microsoft .NET Micro Framework, TinyOS y Linux 2.6.x entre otras. La aplicación desarrollada para este estudio se ha hecho en C# para Microsoft .NET Micro Framework, ya que es una de las configuraciones por la que más se está apostando actualmente.

Una vez elegidas las motas y desplegada la red inalámbrica de sensores, los objetivos que se han buscado en la implementación de este escenario de pruebas, han sido:

- Desplegar un soporte criptográfico basado en AES de 256 bits sobre una WSN.
- Evaluar el incremento del tiempo de cómputo y de consumo energético que conlleva la inclusión de dicho soporte criptográfico.

IV. RESULTADOS

Como ya se ha indicado en apartados anteriores, el objetivo principal de este artículo es presentar los resultados del estudio [9] que se ha realizado con objeto de cuantificar en términos de consumo energético y de tiempo de utilización de recursos de procesamiento lo que implica introducir un criptosistema de calve simétrica basado en AES, en una red inalámbrica de sensores.

Para la realización de las principales mediciones se han modificado las aplicaciones originales, en las cuales en un funcionamiento normal, la mota estaría casi permanentemente dormida y sólo actuaría en caso de producirse alguna variación en sus sensores. Para poder realizar una estimación del consumo de la mota en las peores condiciones posibles, en la aplicación realizada se obliga a la mota a estar continuamente trabajando, es decir, leer de los sensores, preparar el paquete a enviar y mandarlo. Tras un ciclo, se pone a la mota en espera durante unos milisegundos y se vuelve a reanudar el proceso.

A. Medidas del tiempo de cifrado

En primer lugar se realizaron las mediciones de los tiempos de utilización de los recursos de procesamiento durante las fases de cifrado/descifrado, para ello se confeccionó una aplicación que en términos generales arranca un contador al iniciar el programa, detiene el contador una vez se producen todas las operaciones y manda los datos por medio del puerto USB a una aplicación en el ordenador de procesamiento que se encarga de recoger los datos, prepararlos y almacenarlos para poder ser posteriormente tratados con Matlab y de esa forma obtener los parámetros estadísticos deseados.

Se tomaron 3000 muestras de tiempo para ambos procesos. Otros valores de interés que se midieron fueron: media cifrado, 32.4563 ms., varianza cifrado, 0.5183ms., media sin cifrar: 6.8203ms., varianza sin cifrar: 1.2651ms.

Según los resultados obtenidos se observa que en valor medio, la mota tarda aproximadamente cinco veces más en realizar el proceso si la información se envía cifrada.

B. Medidas del consumo energético de las motas

En el estudio se ha realizado un proceso para obtener el consumo de la mota. Este proceso además ha permitido constatar que las medidas del tiempo de cifrado que se presentan en el subapartado anterior son correctas, ya que también se pueden observar los ciclos de tiempo de cifrado gracias a estas mediciones.

Debido a la necesidad de realizar mediciones muy precisas, puesto que se obtuvieron corrientes del orden de miliamperios, y a la escasa duración de los ciclos (del orden de milisegundos) se utilizó una tarjeta de adquisición de datos de National Instruments, modelo CB-68LP.

Para trabajar con este tipo de dispositivos se ha utilizado LabVIEW, generando un VI (que es el nombre que reciben los programas de LabVIEW). En él se realiza un muestreo de los valores de la corriente que consume la mota a una frecuencia de 10 KHz durante 10 segundos, capturando un total de 100.000 muestras.

Debido a que la entrada de la tarjeta de adquisición de datos era en tensión, pero las medidas que se querían realizar eran de corriente, fue necesario adaptar la señal, para de ese modo poder trabajar con ella. Con este fin, se implementó un amplificador de instrumentación, al que por medio de una resistencia de 2 ohmios situada a su entrada, realizaba la conversión entre tensión y corriente.

Como fuente de alimentación se utilizó la HP E3631A que posee una doble salida de tensión. Este modelo de fuente incorpora dos salidas independientes, una capaz de generar entre 0 y 6 voltios y otra salida simétrica que es capaz de dar hasta 25 voltios. El circuito del amplificador de instrumentación está alimentado por medio de la fuente simétrica con una tensión de +15 y -15 voltios. Como la mota está diseñada para funcionar con 3 pilas de 1.5 voltios, se alimentó con una tensión de 4.5 voltios.

La configuración por defecto de la mota en todas las medidas es la que se muestra a continuación:

- El LED que incluye el nodo se encontraba encendido.
- La radio está configurada para emitir con una potencia de -10 dbm en su canal 11.
- La tarjeta de expansión que incorpora los sensores está conectada.

Los resultados obtenidos con las mediciones fueron los siguientes:

El valor medio es de 2,4315 voltios, lo cual supone un consumo medio de corriente de 84 miliamperios. Si alimentamos la mota con 3 pilas AAA de 1.5 voltios cada una y una corriente de 800mAH, el tiempo de vida aproximado de las pilas sería de unas 11 horas.

La máxima corriente consumida se encuentra al inicio del proceso de cifrado y es de 115 mA. El mínimo se encuentra a 50 mA, este valor se produce cuando la mota permanece en espera durante 20 ms tras enviar el paquete.

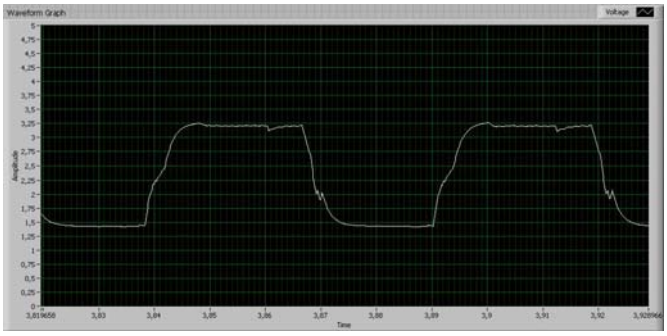


Fig. 1. Consumo del proceso con cifrado.

En la figura 1 se puede observar en detalle el consumo del proceso de cifrado. En el eje de abscisas se encuentra representado el tiempo en segundos y en el de ordenadas, la tensión en voltios.

Al comienzo se ve como la mota se encuentra inactiva, cuando detecta un evento (en este caso se la fuerza a ello), comienza la lectura de los sensores y a preparar el paquete. La recta que aparece a continuación se debe al proceso de cifrado, en el cual el procesador de la mota se encuentra trabajando a pleno rendimiento. Una vez finalizado el proceso de cifrado, la pendiente cae de forma abrupta, ya que el procesador deja de actuar, y aparecen unos picos. Estos picos son debidos al envío del paquete por radio.

Pese a lo que cabía esperar, el consumo del envío por radio no ha resultado tan significativo ya que su duración con respecto a la del proceso de cifrado puede considerarse prácticamente despreciable y su consumo en netamente inferior a éste.

Se realizaron también pruebas comparativas con la radio configurada a máxima potencia (0 dbm) y configurada a la mínima potencia (-25 dbm). En términos de consumo supone una diferencia de aproximadamente 17 mA. Además se observó que el hecho de tener configurada la radio a plena potencia supone un incremento en el consumo general de 5 mA, dato a tener en cuenta en el cálculo del consumo energético de la mota.

Otro de los elementos sobre el que se quiso determinar su consumo fue el del LED que viene incorporado en la placa que supone una diferencia de 50 mV, o 1,7 mA.

Otro de los factores sobre el que interesaba conocer su consumo es la tarjeta de expansión que incluye los sensores. El hecho de desconectar la tarjeta de expansión que incluye los sensores supone reducir el consumo en unos 15 mA.

A continuación se detallan las medidas realizadas cuando la aplicación no realiza el proceso de cifrado de la información antes de enviarla.

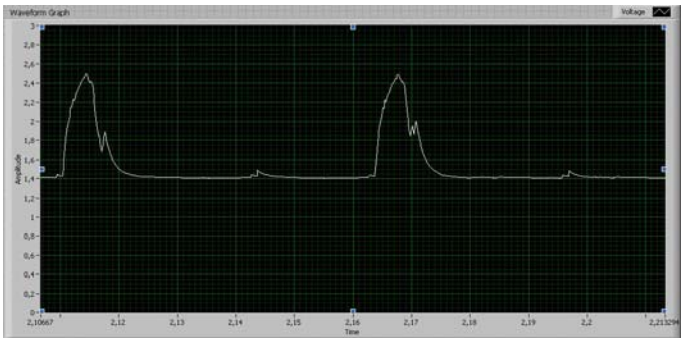


Fig. 2. Consumo del proceso sin cifrado.

El valor medio es de 1,5312 voltios, lo cual supone un consumo medio de corriente de 53 miliamperios. El tiempo de vida aproximado de las pilas sería de unas 20 horas, lo cual supone casi el doble de tiempo de duración que si se realiza el proceso de cifrado.

Como se puede observar en la figura 2, la duración aproximada del proceso sin cifrado es de 8 milisegundos, por lo que el consumo debido al envío por radio supone aproximadamente una tercera parte del tiempo que invierte la mota en realizar las operaciones.

V. CONCLUSIONES

El estudio demuestra que las motas de última generación disponen de suficientes recursos hardware para que las aplicaciones que éstas ejecutan puedan proporcionar servicios básicos de seguridad.

Las plataformas software de desarrollo y de ejecución son de alto nivel, por lo que se facilita el desarrollo de aplicaciones para WSN.

Sin embargo, la configuración elegida para la realización de este estudio arroja unos resultados poco esperados y poco deseados al cifrar los datos con AES-256 bits: el tiempo de vida de la batería se reduce a la mitad, el tiempo de ejecución es cinco veces mayor y el consumo de energía por transmisión de datos es menor que por cómputo de datos.

Este estudio ha dado pie a que se inicien nuevos trabajos en los que se está valorando el efecto de otras variables, como la plataforma de ejecución y la frecuencia de trabajo de la mota, en las prestaciones de aplicaciones protegidas para motas.

AGRADECIMIENTOS

Los trabajos descritos en este artículo se han enmarcado en los proyectos uSWN (*Solving Major Problems in MicroSensorial Wireless Networks*) FP VI y ESNA (*European Sensor Network Architecture*) programa ITEA-Plan Avanza I+D-TSI-020400-2008-127.

REFERENCIAS

[1] F. Anjum and P. Mouchtaris, “Security for Wireless Ad Hoc Networks”. Wiley-Interscience. 2007.

[2] Federal Information Processing Standards, “Specification for the Advanced Encryption Standard (AES)”. FIPS PUB 197 November 2001.

[3] M. Ma, M.K. Denko and Y. Zhang, “Wireless Quality of Service”. CRC Press. 2008.

[4] A.B. García, J.F. Martínez, J.M. López, A. Prayati, “ Problem Solving for Wireless Sensor Networks”. Springer-Verlag. 2008

[5] M. Kuorilehto, M. Kohvakka, J. Suhonen, P. Hämäläinen, M. Hännikäinen and T.D. Hämäläinen, “Ultra-low Energy Wireless Sensor Networks in Practice: Theory, Realization and Deployment”. Wiley-Interscience. 2007.

[6] A. Mishra, “Security and Quality of Service in Ad Hoc Wireless Networks”. Cambridge University Press. 2008.

[7] G. Guimaraes, E. Souto, D. Sadok and J. Kelner, “Evaluarion of Security Mechanisms in Wireless Sensor Networks”. Proceedings of the 2005 Systems Communications (ICW’05) 0-7695-2422-2/05. IEEE 2005.

[8] J.P. Kaps and B. Sunar, “Energy comparison of AES and SHA-1 for ubiquitous computing” Lecture Notes and Computing Sciencien (LNCS). 2006.

[9] A. Maján, “Estudio de Módulos de Seguridad para Redes de Sensores Inalámbricos”. Proyecto Fin de Carrera, EUIT Telecomunicación UPM. 2008.

INFLUENCIA DE LOS PARÁMETROS TOPOLÓGICOS EN LA SINCRONIZACIÓN TEMPORAL DE UNA RED INALÁMBRICA DE SENSORES

José A. Sánchez Fernández, Ana B. García Hernando, José F. Martínez Ortega, Lourdes López Santidrián
Departamento de Ingeniería y Arquitecturas Telemáticas
EUIT de Telecomunicación. Universidad Politécnica de Madrid
Campus Sur UPM. Cta. de Valencia, km. 7. 28031 Madrid
{jsanchez, abgarcia, jfmartin, llopez}@diatel.upm.es

***Resumen*- La Dinámica de Sistemas y la Teoría Algebraica de Grafos proporcionan el marco matemático idóneo para describir analíticamente la evolución temporal de redes complejas, así como para expresar formalmente su topología, cuestiones particularmente útiles en las redes inalámbricas de sensores (WSN). La combinación de las herramientas matemáticas que proporcionan estas teorías permite analizar detalladamente diversas características y propiedades de las WSN, entre ellas la capacidad de sincronización temporal de sus nodos, un aspecto de gran importancia en numerosas aplicaciones. En esta comunicación se aplican algunas de estas herramientas para determinar la capacidad de sincronización de una WSN diseñada específicamente para dar soporte a aplicaciones de vigilancia perimetral.**

***Palabras Clave*- red inalámbrica de sensores; protocolo de sincronización; teoría espectral de grafos; aplicación de vigilancia perimetral.**

I. INTRODUCCIÓN

El formalismo matemático establecido para el estudio de redes inalámbricas de sensores (WSN) está inspirado en los modelos desarrollados para sistemas complejos, constituidos por un conjunto de osciladores mutuamente acoplados. El análisis de estos sistemas revela ciertas analogías con respecto a las WSN, puesto que un sensor, como nodo de la red, puede asociarse con un oscilador que interacciona con su entorno físico y con el resto de nodos. Se han propuesto diversos modelos matemáticos para describir la sincronización de los nodos de redes complejas. Mirollo y Strogatz [1] estudiaron la sincronización de sistemas biológicos compuestos de osciladores acoplados idénticos. Los sistemas químicos fueron estudiados por Kuramoto [2], que propuso un modelo dinámico no lineal para osciladores fuertemente acoplados en fase. Barahona y Pecora [3, 4] establecieron un modelo lineal para estudiar la sincronización de redes formadas por osciladores idénticos, y Barbarossa, Celano y Scutari [5, 6] propusieron una extensión del modelo de Kuramoto aplicable a las WSN.

La Dinámica de Sistemas proporciona una descripción analítica de la evolución temporal de las variables de estado que caracterizan a cada uno de los nodos. Esta evolución, así como la interacción o acoplamiento de cada nodo con el resto de la red, se describe mediante un sistema de ecuaciones diferenciales. La información sobre la topología lógica de la red (la existencia de enlaces de comunicación entre cada par de nodos) está contenida en su grafo asociado, que se representa mediante una serie de matrices, aplicándose para su estudio la Teoría Algebraica de Grafos [7]. Los resultados obtenidos muestran que las condiciones de sincronización de una WSN descansan sobre varias inecuaciones que relacionan diversos parámetros ligados a la topología de la red [4, 8]. El principal objetivo de esta comunicación es aplicar algunos de estos resultados a una WSN diseñada como soporte de aplicaciones de vigilancia perimetral, en el ámbito de un proyecto de I+D del VI Programa Marco de la Unión Europea (UE).

El resto de esta comunicación se organiza de la siguiente forma: la Sección II introducirá el formalismo matemático que describe la evolución de una red compuesta de osciladores mutuamente acoplados; la Sección III mostrará los resultados que establecen las condiciones que permiten alcanzar la sincronización de la red; la Sección IV presentará las características de la WSN de vigilancia perimetral y su dominio de aplicación, justificando la necesidad de conseguir la sincronización de sus nodos; por último, la Sección V discutirá los resultados de aplicar el modelo matemático para establecer las condiciones de sincronización de la WSN.

II. FORMALISMO MATEMÁTICO

De entre los modelos matemáticos planteados, se ha seleccionado por su generalidad el desarrollo efectuado por Barahona y Pecora [3, 4], que considera una red genérica compuesta de N nodos, cada uno de ellos asimilable a un sensor que interacciona con su entorno físico. El i-ésimo sensor ($i = 1,\ldots,N$) obtiene, a lo largo del tiempo, medidas de

M magnitudes físicas (temperatura, presión, movimiento, etc.), representadas mediante un vector de variables de estado, $\boldsymbol{x}_i(t) = (x_{i1}(t),\ldots,x_{iM}(t))$. Los sensores actúan como osciladores mutuamente acoplados al intercambiar entre sí algunos de los datos recopilados o determinadas marcas de sincronización. Así, cada nodo puede adaptar la evolución de sus variables de estado de acuerdo con los datos recibidos del resto, como por ejemplo los valores del tiempo de sus relojes locales para coordinar la sincronización.

El sistema dinámico presente en cada nodo evoluciona de acuerdo con el siguiente sistema de ecuaciones diferenciales de primer orden, según se propone en [4]:

$$\frac{dx_i}{dt} = F(x_i(t)) - \sigma\sum_{j=1}^{N} L_{ij}H(x_j(t)), i = 1,\ldots, N \qquad (1)$$

donde $\boldsymbol{F}(x_i(t))$ es una función vectorial de dimensión *M*, que expresa la evolución de las variables de estado $\boldsymbol{x}_i(t)$de cada nodo, y σ es un factor de acoplamiento global, que se supone idéntico en todos los nodos de la red. La función vectorial de las variables de estado de cada oscilador, $\boldsymbol{H}(x_j(t))$, de dimensión *M*, representa el acoplamiento selectivo entre los osciladores presentes en los nodos. Los elementos L_{ij} son las componentes de una matriz **L**, de dimensiones $N \times N$, que especifica cuáles de los nodos están conectados entre sí mediante un enlace de comunicación y cuáles no.

L es una matriz simétrica, ya que la red no dispone de enlaces direccionales entre sus nodos. Los resultados obtenidos en [4] establecen las condiciones genéricas que deben satisfacer las funciones $\boldsymbol{F}(x_j(t))$ y $\boldsymbol{H}(x_j(t))$ y la matriz **L** para alcanzar un estado de sincronización estable en la red de osciladores acoplados cuya dinámica viene determinada por (1). En particular, la determinación de los elementos L_{ij} de L será crucial para estudiar la sincronización de la red.

Para facilitar el análisis, (1) puede simplificarse adecuadamente. Puesto que nuestro principal interés es la sincronización temporal de los nodos de la red, se considera únicamente una variable de estado $x_i(t)$ en cada nodo, correspondiente al tiempo local medido por éste a partir de su oscilador *hardware* interno. En este caso, (1) se convierte en

$$\frac{dx_i}{dt} = \omega - \sigma\sum_{j=1}^{N} L_{ij}H(x_j(t)), i = 1,\ldots, N \qquad (2)$$

donde $\boldsymbol{\omega}$ representa la frecuencia de oscilación de los relojes de los nodos, que se supone idéntica y constante en todos ellos, ya que disponen del mismo oscilador *hardware*. Si no existe acoplamiento entre los nodos, puede eliminarse la suma del segundo miembro de (2), obteniéndose

$$x(t) = x(0) + K\int_0^t \omega(\tau)d\tau \qquad (3)$$

que es la expresión usual del tiempo medido por un reloj $x(t)$, a partir de un oscilador *hardware* de frecuencia angular $\boldsymbol{\omega}$. Una vez sincronizada la red, (2) debe ser equivalente en todos sus nodos, lo que implica que la suma de su segundo término es constante:

$$\sum_{j=1}^{N} L_{ij}H(x_j(t)) = Cte \qquad (4)$$

Sin perder generalidad, puede hacerse que $Cte = 0$, lo que hace que la suma de los elementos de las filas de **L** sean 0:

$$\sum_{j=1}^{N} L_{ij} = 0 \qquad (5)$$

Estas restricciones permiten determinar los elementos L_{ij}, aunque es necesario introducir antes algunos conceptos básicos de Teoría Algebraica de Grafos. Una red puede modelarse mediante un grafo $U=U(V,E)$, compuesto de *N* nodos o vértices *V*, etiquetados de 1 a *N*, y un conjunto de conexiones o enlaces *E* entre ellos. El número de enlaces puede variar entre 0 (no existen conexiones entre los nodos) y $N(N-1)/2$ (cada nodo está conectado con el resto). Un grafo puede representarse inicialmente mediante su matriz de adyacencia **A**, una matriz simétrica de dimensiones $N \times N$, donde $A_{ij} = 1$ si los nodos *i* y *j* están conectados, y $A_{ij} = 0$ en caso contrario. Las componentes de la diagonal principal de **A** se definen como $A_{ii} = 0$. Los invariantes de la matriz de adyacencia **A** bajo permutaciones de las etiquetas asociadas a los nodos (en particular, las propiedades de su espectro) han sido ampliamente estudiados [4, 7, 8].

El grado d_i del nodo *i* es la suma del número de enlaces que lo conectan a otros nodos, que puede obtenerse a partir de la suma de los elementos de la fila *i*-ésima de **A**:

$$\sum_{j=1}^{N} A_{ij} = d_i \qquad (6)$$

Esto sugiere formar una nueva matriz **D** (matriz de valencia). Los elementos de su diagonal principal, D_{ii}, son equivalentes a las sumas *i*-ésimas de (6), siendo el resto de elementos iguales a 0. A partir de **A** y **D** se define finalmente una nueva matriz **L** (matriz laplaciana), como

$$L_{ij} = \delta_{ij}D_{ij} - A_{ij} \qquad (7)$$

Esta matriz satisface los requisitos de sincronización establecidos en (5). Las propiedades espectrales de la matriz laplaciana **L** también se han estudiado, lo que evita tener que resolver su ecuación característica, $det(\mathbf{L} - \gamma\mathbf{1}) = 0$, para obtener sus valores propios. En la Sección IV, (12) muestra un ejemplo de matrices de incidencia y laplaciana asociadas a una WSN con topología en anillo. En la siguiente Sección se estudiarán las condiciones generales de sincronización de una red, a partir del análisis de los valores propios de **L**.

III. Condiciones Genéricas de Sincronización en WSN

A continuación se mostrarán algunos resultados presentados en [4] y [8], útiles desde el punto de vista de la sincronización. **L** es una matriz semidefinida positiva, con un número máximo de *N* valores propios, $\gamma_1 \leq \gamma_2 \leq \ldots \leq \gamma_N$. Por construcción de **L**, su menor valor propio, γ_1, es 0. El siguiente, γ_2, se denomina conectividad algebraica, y juega un importante papel en la sincronización de la red. Un valor grande de γ_2 está asociado a una red con buena conectividad entre sus nodos, lo que facilita su sincronización. De otro modo, si $\gamma_2 \rightarrow 0$, la red no podrá alcanzar una sincronización adecuada. Se ha obtenido que el tiempo en alcanzarse la sincronización, para el caso especificado en (2), es proporcional a γ_2^{-1} [9]. Aunque γ_2 no proporciona

información sobre la topología de la red, se han obtenido cotas para él que incorporan diversos parámetros topológicos, presentes en las siguientes inecuaciones [4, 8]:

$$\gamma_2 \le \frac{N}{N-1}\delta \le \frac{N}{N-1}\Delta \le \gamma_N \le 2\Delta \quad (8)$$

$$\frac{4}{ND} \le \gamma_2 \le \frac{Nd_i}{N-1}, \forall i \quad (9)$$

Aquí, d_i representa el grado del nodo i, δ es el grado mínimo del grafo (el valor mínimo de los grados de los nodos) y Δ es el grado máximo del grafo (el valor máximo de los grados de los nodos). La distancia entre dos nodos se define como el menor número de enlaces que hay que atravesar para ir de uno al otro. El diámetro del grafo, D, es el valor máximo de estas distancias. Las redes con valores grandes de N y D proporcionan una cota inferior pequeña para γ_2. Sin embargo, si estos valores son menores, γ_2 será mayor y la red se sincronizará con mayor facilidad.

Además, el cociente γ_N/γ_2 debe ser lo menor posible. El valor más pequeño es 1, únicamente posible si cada nodo está conectado con el resto. A partir de (8) y (9), se obtiene

$$\frac{\Delta}{\delta} \le \frac{\gamma_N}{\gamma_2} \quad (10)$$

Si el cociente γ_N/γ_2 proporciona un valor grande (existe gran diferencia entre los grados máximo y mínimo del grafo), la sincronización será difícil. Se ha obtenido una cota máxima para γ_N/γ_2 [8]:

$$\frac{\gamma_N}{\gamma_2} \le \frac{ND\Delta}{2} \quad (11)$$

Así, si se reducen el grado máximo Δ, el diámetro D o el número de nodos N, se verá mejorada la sincronización de la red asociada. Por tanto, la Dinámica de la red y la Teoría Algebraica de Grafos proporcionan herramientas útiles para modelar una red y obtener sus condiciones de sincronización, que descansan en un conjunto de parámetros topológicos globales. La siguiente sección describe el escenario de despliegue de la aplicación de vigilancia perimetral a que se ha hecho referencia, justificando además la necesidad de adoptar un esquema de sincronización temporal en dicho escenario.

IV. Dominio de Aplicación Específico: WSN de Vigilancia Perimetral

Algunas de las aplicaciones que se benefician más del uso de una WSN son las de vigilancia perimetral de edificios o zonas específicas. Parte del trabajo desarrollado en el proyecto µSWN (Solving Major Problems in Microsensorial Wireless Networks) [10], financiado por el VI Programa Marco de la UE, ha consistido en el diseño de una aplicación de vigilancia sobre un perímetro virtual cerrado. Esta aplicación utiliza una WSN, que dispone de dos tipos de agentes que se ejecutan en sus motas o nodos, denominados agentes perimetrales y de pulsera. Las motas perimetrales se despliegan a lo largo de un perímetro virtual cerrado (véase la Fig. 1), configurando una sencilla topología lógica en anillo.

Para visualizar esta topología, y según lo expuesto en la Sección II, se muestran a continuación las matrices de adyacencia **A** y laplaciana **L** asociadas, suponiendo enlaces de comunicación sólo entre nodos adyacentes y con $N = 5$ nodos:

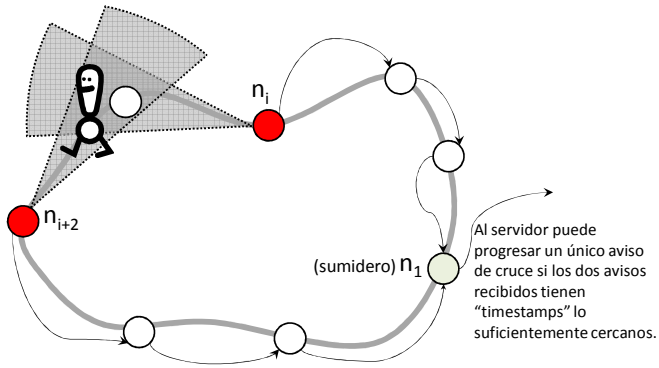


Fig. 1. Dos alarmas (nodos n_i y n_{i+2}) asociadas al mismo evento de cruce.

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} \quad L = \begin{pmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix} \quad (12)$$

Las motas perimetrales disponen de uno o dos sensores infrarrojos pasivos de presencia (PIR). Cada segmento perimetral limitado por dos motas vecinas está cubierto al menos por uno de estos sensores. Si una persona o vehículo cruza el perímetro, uno o dos sensores PIR lo detectan y activan el correspondiente agente perimetral (véase la Fig. 1). Éste trata de averiguar si ha sido causado por una persona autorizada (que porta una pulsera que incorpora una mota con un agente de pulsera) o por un intruso. Para ello difunde un mensaje de solicitud de identificación dentro de su radio de cobertura. Un agente de pulsera que recibe esta solicitud responde con su identificador, que será posteriormente retransmitido al nodo sumidero. Si el agente perimetral no recibe respuesta en un tiempo razonable, asume que un intruso ha cruzado el perímetro y lo notifica al sumidero. En este escenario el sumidero puede recibir varias notificaciones correspondientes a un único cruce con una diferencia temporal significativa entre ellas debido a lo siguiente:

- Los tiempos necesarios para alcanzar el sumidero por paquetes generados simultáneamente en distintos nodos pueden ser altos y distintos. Ya que es importante limitar el consumo energético en una WSN, numerosos protocolos de comunicación implementan ciclos de baja actividad en los que los nodos se activan y retransmiten información durante una pequeña fracción del tiempo asignado al ciclo. Si el paquete debe atravesar un número elevado de nodos, el retardo total puede ser considerable.
- Dependiendo de la relación entre el radio de cobertura de las motas y el área de detección de los sensores PIR, es posible que dos ó más nodos detecten el mismo evento de cruce e informen de ello al sumidero (véase la Fig. 1). También es posible que algunas de estas notificaciones sean del tipo "autorizado" y otras del tipo "intruso" a pesar de corresponder al mismo cruce, debido a deficiencias de cobertura en los nodos que impiden que llegue la respuesta del agente de pulsera a todos los agentes perimetrales que detectaron dicho cruce.

Si los mensajes de aviso contuvieran las marcas de tiempo locales de las motas que los originaron (el momento en el que el agente perimetral detectó el cruce), sería más sencillo localizar y eliminar la información duplicada. Esto será posible si los nodos disponen de relojes sincronizados.

Aunque el rango de detección de los sensores PIR no suele ser configurable (lo que dificulta evitar detecciones duplicadas en algunos escenarios de despliegue), el radio de cobertura de un nodo sí suele serlo. Esto implica que la topología lógica de la WSN puede variarse ajustando la potencia de transmisión de los nodos, limitando así el número de vecinos de cada mota.

Una vez justificada la necesidad de sincronización de los nodos de la WSN, hay que escoger un esquema de sincronización adecuado a este dominio de aplicación. Se han planteado diversas propuestas de protocolos de sincronización [11]. La sencillez de la topología de la red permite optar por un esquema de sincronización global mediante la difusión periódica de un mensaje que contiene una marca de tiempo maestro, t_M, que partirá del sumidero n_1 hacia el resto de nodos, esquema similar al protocolo de sincronización propuesto en [12]. El mensaje de difusión de t_M recorre el siguiente trayecto secuencialmente:

$$n_1 \rightarrow n_2 \rightarrow n_3 \rightarrow \ldots \rightarrow n_N \rightarrow n_1$$

Cada nodo registra y retransmite el tiempo maestro t_M hasta completar el circuito, así como la marca de tiempo local t_i asociada al instante de recepción del mensaje que contiene t_M. Una vez completado el trayecto de este mensaje, n_1 envía un nuevo mensaje en secuencia, que contiene la marca de tiempo del instante de recepción del menaje inicial de sincronización, t_E. Como cada nodo n_i conoce su posición en la secuencia de retransmisión de ambos mensajes, es posible ajustar el tiempo de su reloj local, t, a partir de la expresión:

$$t \leftarrow t - (t_i - t_M) + \frac{t_E - t_M}{N}(i-1) \qquad (13)$$

Los nodos no deben retrasar sus relojes locales, para evitar problemas de relación causal entre eventos registrados en instantes de tiempo próximos. La siguiente Sección aplicará los resultados de la Sección III para determinar la topología lógica idónea de la WSN de vigilancia perimetral, a efectos de favorecer y simplificar la sincronización de sus nodos.

V. Determinación de las Condiciones de Sincronización de la WSN de Vigilancia Perimetral

En la Sección III se han mostrado los resultados matemáticos que relacionan la capacidad de sincronización de una red con diversos parámetros topológicos globales. Aplicando (10) y (11) a la WSN de vigilancia perimetral descrita, se obtienen las siguientes conclusiones:

- El número de nodos, N, debe reducirse lo más posible, pero asegurando la conectividad entre cada nodo n_i y sus vecinos n_{i-1}, n_{i+1}, con objeto de mantener cerrado el perímetro de vigilancia.
- El diámetro de la WSN, D, debe reducirse también. Si se incrementa el radio de cobertura de cada nodo n_i para tener conectividad con los nodos n_{i-2} e n_{i+2}, D se reducirá en un factor de ½. Mayores incrementos del radio de cobertura reducirán D aún más.
- El grado máximo de los nodos, Δ, no debe incrementarse demasiado: debe existir un equilibrio entre la reducción del diámetro D y el correspondiente incremento de Δ.
- La dispersión entre los grados mínimo (δ) y máximo (Δ) de la WSN debe ser lo más pequeña posible, lo que puede conseguirse fijando una única distancia física entre nodos adyacentes y un único radio de cobertura, para que se cumpla la igualdad $\delta = \Delta$.
- Con respecto al protocolo de comunicación, la sincronización será tanto más exacta cuanto menor sea el tiempo de procesamiento y transmisión de mensajes con respecto al ritmo de actualización de los relojes locales de los nodos a partir de sus osciladores.

VI. Conclusiones

Esta comunicación ha aplicado resultados procedentes de la Dinámica de Sistemas y de la Teoría Algebraica de Grafos para determinar las condiciones de sincronización de una WSN de vigilancia perimetral con topología en anillo. Las relaciones matemáticas entre los valores propios de la matriz laplaciana del grafo que modela la red y varios parámetros topológicos globales a ésta permiten determinar la capacidad de sincronización de la WSN.

La sincronización se beneficiará de una reducción del número de nodos N, de su diámetro D, del grado máximo de la red Δ y de una distribución equidistante de los nodos, con un radio de cobertura equivalente en todos ellos, que facilite la equivalencia entre los grados mínimo δ y máximo Δ. El trabajo a desarrollar en el futuro consistirá en verificar las condiciones de sincronización enunciadas anteriormente en el escenario de aplicación descrito en esta comunicación. Se espera que los resultados obtenidos puedan guiar el diseño de WSN dedicadas a vigilancia perimetral y poder extraer reglas más generales para WSN diseñadas con otros propósitos.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el VI Programa Marco de la UE, a través del proyecto µSWN (Solving Major Problems in Microsensorial Wireless Networks), código IST-034642.

Referencias

[1] R. E. Mirollo y S. Strogatz, “Synchronization of pulse-coupled biological oscillators”, SIAM Journal of Applied Mathematics., vol. 50, no. 6, pp.1645-1662, Diciembre 1990.
[2] Y. Kuramoto, Chemical Oscillations, Waves and Turbulences, Springer, 1984.
[3] M. Barahona y L. M. Pecora, “Synchronization in Small-World sytems”, Physical Review Letters, vol. 89, no. 5, pp. 054101-1-054101-4, Julio 2002.
[4] L. M. Pecora y M. Barahona, “Synchronization of Oscillators in Complex Networks”, Chaos and Complexity Letters, vol. 1, no. 61, pp. 61-91, 2005.
[5] S. Barbarossa y F. Celano, “Self-organizing sensor networks designed as a population of mutually coupled oscillators”, Proc. of IEEE SPAWC’05, New York, USA, 2005.
[6] S. Barbarossa y G. Scutari, “Decentralized Maximun-Likelihood Estimation for Sensor Networks Composed of Nonlinearly Coupled Dymanical Systems”, IEEE Transactions on Signal Processing, vol. 55, no. 7, pp. 3456-3470, Julio 2007.
[7] N. Biggs, Algebraic Graph Theory. Cambridge University Press, 1974.
[8] F. Comellas y S. Gago, “Synchronizability of complex networks”, Journal of Physics A: Mathematical and Theoretical, vol. 40, pp. 4483-4492, IOP Publishing, 2007.
[9] J. A. Almendral y A. Díaz-Guilera, “Dynamical and spectral properties of complex networks”, New Journal of Physics, 9 (2007) 187, doi: 10.1088/1367-2630/9/6/187.
[10] Página web del proyecto europeo µSWN: http://www.uswn.eu.
[11] B. Sundararaman, U. Buy y A. D. Kshemkalyani, “Clock Synchronization for Wireless Sensor Networks: A Survey”, Ad Hoc Networks, vol. 3, no. 3, pp. 281-323, Mayo 2005.
[12] Q. Li, D. Rus, “Global Clock Synchonization in Sensor Networks”, IEEE Transactions on Computers, vol. 55, no. 2, pp. 214-226, Febrero 2006, doi:10.1109/TC.2006.25.

P2PKEP: Peer-to-Peer Key Exchange Protocol

Juan Carlos Otero, Enrique de la Hoz, Bernardo Alarcos, Iván Marsá-Maestre, Alicia Martínez.
Área de Ingeniería Telemática. Departamento de Automática. Universidad de Alcalá
28871 Alcalá de Henares (Madrid)

{jcotero,enrique.delahoz,bernardo.alarcos,ivan.marsa,alicia.martinez}@uah.es

***Resumen*- En este artículo se presenta P2PKEP, un protocolo que, basándose en certificados de clave pública, permite autenticar a los nodos de una red ad-hoc con parte de infraestructura fija y el intercambio de material criptográfico con el que crear Asociaciones de Seguridad que permitirán que los distintos nodos puedan comunicarse de forma segura en la capa de aplicación. Este protocolo está basado en el *Handshake de TLS* adaptándolo a las redes ad-hoc para que sea más ligero y eficiente. También se verifica la eficiencia de la criptografía de curvas elípticas frente a la criptografía tradicional de cara a su utilización en un entorno de redes ad-hoc.**

***Palabras Clave*- redes ad-hoc, PKI, protocolo de intercambio de claves, Asociación de seguridad, criptografía ECC**

I. INTRODUCCIÓN

En las redes ad-hoc, por su propia naturaleza, no se dispone de infraestructura fija y esto hace que las soluciones clásicas de PKI (*Public Key Infrastructure*) no sean válidas, teniendo que buscar otras soluciones basadas en la cooperación entre los nodos de la red para distribuir servicios que habitualmente son centralizados [1]. Sin embargo, hay determinados entornos de redes ad-hoc que sí cuentan con una parte de infraestructura fija, como pueden ser las redes vehiculares [2], las redes ad-hoc desplegadas como soporte a la intervención en zonas de desastre [3], [4] o las redes personales desplegadas en una ciudad [5]. En este tipo de escenarios, se puede aprovechar dicha infraestructura para implantar una arquitectura PKI para que los nodos de la red ad-hoc cuenten, como punto de partida, con claves públicas adquiridas mediante certificación de manera confiable.

En estos entornos, los nodos se intercambian información de control que ayuda, por ejemplo, a configurar el encaminamiento, pero también se intercambian información de aplicación que tendrá unos requisitos de seguridad distintos y puede requerir que dicha información no sea visible por los nodos intermedios. Debido a esto, es posible que los mecanismos para asegurar la información de la capa de control no sean apropiados para asegurar la información de la capa de aplicación. Por lo tanto, es necesario habilitar mecanismos de seguridad diferentes en las distintas capas de comunicación.

El presente trabajo se centra en proponer un protocolo que, basándose en los certificados de clave pública y en el resto de mecanismos asociados (certificación, revocación, verificación, etc.), se emplee para autenticar a los nodos de la red ad-hoc y para el intercambio de material criptográfico con el que crear las Asociaciones de Seguridad, que serán las que permitirán a las aplicaciones de distintos nodos comunicarse de forma segura.

El resto del artículo se estructura de la siguiente forma. En la sección II se describen los objetivos que perseguimos y los detalles a tener en cuenta para alcanzarlos. Las características principales del protocolo P2PKEP, las fases del protocolo, los mensajes y su significado se presentan en la sección III. En la sección IV se presentan las pruebas que se han realizado. Finalmente, las secciones V y VI se dedican a las conclusiones y a las líneas de investigación futuras respectivamente.

II. P2PKEP. OBJETIVOS Y CARACTERÍSTICAS

El protocolo debe adaptarse a la naturaleza de las redes ad-hoc: nodos con bajos recursos, ancho de banda limitado, comunicaciones peer-to-peer, etc. Por tanto, buscaremos que el protocolo sea eficiente, lo más sencillo posible y robusto, en el sentido de que sólo se permitirá el uso de mecanismos criptográficos seguros y ligeros.

Para conseguir los objetivos anteriores, proponemos el empleo de un mecanismo de autenticación mutua basado en el *Handshake de TLS* (*Transport Layer Security*) [6] con algunas modificaciones encaminadas a adaptar este protocolo para el entorno de las redes ad-hoc. Las principales modificaciones introducidas son las siguientes:

- Como estamos en un escenario ad-hoc, cualquiera de los dos nodos puede iniciar la comunicación, por lo tanto protocolo debe ser p2p (*peer-to-peer*), en el que, inicialmente, no hay un nodo que haga el papel de servidor y otro de cliente. Debido a esto, será obligatorio autenticar a ambos nodos.
- Negociación sencilla de capacidades.
- Uso de criptografía ECC (*Elliptic Curve Cryptography*, ECC) como alternativa para reducir el tiempo de cómputo, la cantidad de espacio de almacenamiento necesaria y el tamaño de los mensajes.
- Envío de mensajes sobre datagramas UDP en vez de sobre TCP, con el objetivo de mejorar la eficiencia.

Las principales características del protocolo P2PKE se describen en los siguientes puntos.

A. Comunicación fiable.

El protocolo P2PKEP está constituido por un número pequeño de mensajes de reducido tamaño, por lo que proponemos que los mensajes se intercambien sobre UDP en vez de sobre TCP como lo hace TLS.

El trabajar usando UDP como protocolo de transporte, implica que tenemos que ofrecer fiabilidad en los mensajes intercambiados, para ello hemos utilizado identificación de los mensajes, máquinas de estado y temporizadores.

B. Uso de certificados digitales

Si aplicamos este protocolo en un escenario de rescate como el descrito en [3], se puede desplegar una infraestructura PKI con dos niveles jerárquicos de CAs. En este escenario, será necesario pedir un solo certificado (el del nodo con el que se quiere comunicar) en la mayoría de los casos y, con menor frecuencia, el del nodo y el de su CA. Gracias a esto, podemos optimizar el protocolo y enviar únicamente los certificados que se necesitan para validar el camino de certificación.

En cuanto al tipo de clave que contienen los certificados de los nodos, proponemos el uso de criptografía asimétrica basada en curvas elípticas (ECC) como opción, por el reducido tamaño de la clave con respecto a RSA y por su mayor eficiencia en generar firmas digitales.

Debido a que la operación de verificación de la firma es mucho más rápida en RSA ([7,8]), proponemos el uso de certificados híbridos (con claves ECC pero firmados con RSA). De esta forma, la firma de los certificados (que es la parte costosa) es realizada por la CA una única vez, mientras que la verificación (que es mucho más rápida) la deben realizar los nodos muchas veces. En la sección de pruebas compararemos estas opciones con el objetivo de valorar la eficiencia de utilizar una u otra.

Podemos considerar este escenario similar al caso de una red VANET, en la cual los vehículos tuvieran certificados y existiera una arquitectura de PKI jerárquica organizadas por ejemplo por zonas geográficas o por autoridades de tráfico [9,10].

C. Fragmentación

Para evitar los costes derivados de implementar una capa de fragmentación y ensamblado, hemos evitado la fragmentación enviando mensajes del tamaño adecuado para que la información se intercambie por el medio compartido sin fragmentar y de forma eficiente. Según el estudio realizado en [11], el tamaño de los mensajes en redes ad-hoc que permite una comunicación más eficiente se encuentra entre 250 bytes y 750 bytes, siendo 500 bytes el tamaño óptimo para una tasa de error de bit (BER) de 10^{-4}.

En casos extremos (uso de RSA con clave de 3072 bits e intercambio de parámetros de *Diffie-Helman Ephemeral* de 1024 bits) podemos llegar a mensajes entorno a los 1000 bytes.

III. Mensajes y fases del protocolo

En esta sección se describen las características de los mensajes y fases del protocolo propuesto, y sus diferencias con el *Handsake de TLS*.

A. Inicio del protocolo y asignación de roles

En las redes p2p, todos los nodos tienen igual jerarquía de modo que cuando dos nodos se encuentran en el alcance en la red, cualquiera de ellos puede iniciar las comunicaciones con el otro. En P2PKEP, al nodo que inicia el protocolo lo llamamos *Initiator*, y al que responde a la petición lo llamamos *Responder*.

Cada vez que un nodo descubre a otro en la red, si quiere comunicarse de forma segura con él, debe establecer una asociación de seguridad entre ambos, si es que no había una establecida previamente. Para ello, se empleará el protocolo P2PKEP. El protocolo se inicia con el envío de un mensaje HELLO hacia el otro nodo. Durante la fase de inicio, se pueden dar las siguientes situaciones:

1. Un nodo inicia la conversación enviando I_HELLO al otro nodo y actuando, por tanto, como *Initiator*. El otro extremo le responde con R_HELLO actuando como *Responder*.
2. Ambos nodos se descubren y, dentro de una ventana de tiempo reducida, intentan iniciar el protocolo enviando el mensaje I_HELLO (i.e., un nodo envía I_HELLO hacia el otro y recibe I_HELLO del otro nodo en lugar R_HELLO). Uno de ellos debe desistir de su intento y actuar como *Responder*. El conflicto se resuelve evaluando las direcciones IP de los nodos en contienda: el nodo cuya dirección IP es menor debe actuar como *Responder* y enviar un R_HELLO al otro extremo. El otro nodo, que será el que actúe como *Initiator*, simplemente ignorará el I_HELLO recibido y esperará el R_HELLO correspondiente de manera normal.

La figura 1 muestra la máquina de estados que resuelve el conflicto de roles durante el inicio del protocolo.

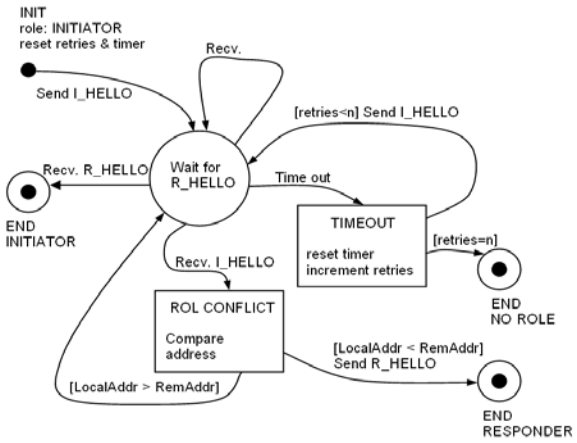


Fig. 1. Máquina de estados para resolver el conflicto de roles.

Si se recibe "OTHER" (i.e., un mensaje inesperado) se ignora para evitar ataques de DoS de otros nodos. Se establecerá un número máximo de reintentos antes de desistir de establecer la asociación.

B. Negociación de capacidades

Se ha simplificado la negociación de capacidades de TLS para conseguir dos objetivos: que la negociación sea más rápida y mejorar la robustez.

Para ello, el nodo que actúa como *Initiator* envía una única lista con los *ciphersuites* que soporta en I_HELLO y el *Responder* informará, en R_HELLO, de cual es el *ciphersuite* más robusto y eficiente que soporta de dicha lista; si no es capaz de soportar ninguno, no se podrá establecer una conexión segura entre ambos. En la lista de *ciphersuites* que propone el *Initiator* no debe haber ninguno débil.

C. Intercambio de información de autenticación

En esta fase los nodos se intercambian los certificados, los parámetros públicos y claves públicas de Diffie-Hellman mediante el mensaje KEY_EXCHANGE. Dado que estamos en un entorno en el que todos los nodos disponen de un certificado, se firmarán los mensajes KEY_EXCHANGE en ambos sentidos para que exista autenticación mutua. Con esta variante DH a la que se denomina *Diffie-Helman Ephemeral* (DHE), se evita el ataque del intermediario. Las opciones que usaremos para el intercambio de clave son DHE y su variante basada en curvas elípticas (ECDHE). Con estos parámetros, ambos nodos, generarán un valor secreto común, denominado *premastersecret* (*pms)*, que será usado como semilla para generar las claves simétricas.

Para que un nodo pueda verificar la autenticidad de la clave pública de otro, necesitará verificar la cadena de certificados hasta su CA de confianza, esta cadena como máximo será de dos certificados, como se ha visto en la sección II.B. Cada nodo enviará en su contenedor HELLO su identidad y la de su CA (Nid|CAid), con esta información los nodos sabrán qué certificados necesitan y enviarán una petición de certificados (CERTIFICATE_REQUEST). Para el intercambio de certificados se enviará un mensaje CERTIFICATE por cada uno de los certificados pedidos.

D. Finalización del protocolo

Los contenedores FINISHED están basados en los mismos mensajes que usa TLS. En la figura 2 podemos ver el diagrama de mensajes que necesitan intercambiar *Initiator* y *Responder* durante la ejecución del protocolo.

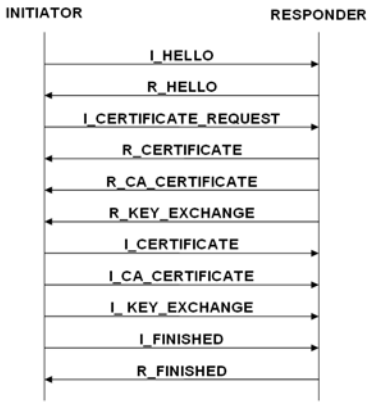


Fig. 2. Mensajes intercambiados entre el *Initiator* y el *Responder*

Los parámetros necesarios para realizar el intercambio de clave dependen de si se ha acordado utilizar DHE o ECDHE. Además, el mensaje R_KEY_EXCHANGE contiene un campo CERT_REQ que le indica al *Initiator* qué certificados debe enviar al *Responder* de forma análoga a como se hacía en el mensaje I_CERTIFICATE_REQUEST.

Los tres siguientes mensajes serían los mensajes correspondientes a los anteriores por parte del *Initiator*.

Los dos últimos mensajes (I_FINISHED y R_FINISHED) sirven para verificar que todo ha ido bien de forma análoga a los mensajes FINISHED de TLS.

IV. PRUEBAS

Se ha realizado una implementación de las funcionalidades más importantes del protocolo propuesto y del protocolo *Handshake de TLS.* Con estas pruebas, pretendemos evaluar el grado de mejora en la eficiencia de P2PKEP con respecto al *Handshake de TLS*, comparando el tiempo en establecerse una conexión con ambos protocolos. Evaluaremos también la mejora introducida por el uso de criptografía basada en curvas elípticas (ECC). Se han realizado dos tipos de medidas:

- Comparación del protocolo *Handshake* de TLS con P2PKEP para determinar la mejora introducida con nuestra propuesta.
- Comparación de cada uno de los métodos de intercambio de claves implementados en su versión empleando criptografía basada en curvas elípticas con la criptografía convencional para analizar las mejoras que pueden introducir el uso de curvas elípticas.

A. Escenario de pruebas

Por motivos de simplicidad, se ha usado una red ad hoc real de dos máquinas con los roles de los nodos prefijados de antemano, de manera que un nodo actúa siempre como *Initiator* y el otro siempre como *Responder*.

Con objeto de simular las capacidades que tendrán los nodos de la red ad-hoc, se han usado ordenadores con pocos recursos, sus características son las siguientes:

- **Responder**: AMD Athlon 64 a 500 MHz, RAM 350MB Ubuntu 2.60.20-15.
- **Initiator**: AMD K6 3D a 450 MHz, RAM 128 MB, Ubuntu 2.6.20-16.

B. Descripción y resultados de las pruebas

Se ha implementado el protocolo P2PKE de forma completa enviando todos los certificados en todos los casos y con preasignación de roles.

En la tabla 1 se muestran los retardos medios obtenidos comparando P2PKEP y TLS, con diferentes combinaciones de *ciphersuite*. Como se puede observar, las mejoras en el tiempo de establecimiento de una conexión conseguidas por el empleo de P2PKE frente a TLS están en torno a un 30%-40% con independencia del método empleado para la negociación de las claves y del uso o no de criptografía de curvas elípticas.

CIPHER SUITE	DHE / ECDHE	FIRMA	TLS (ms)	P2PKEP (ms)	MEJORA (%)
DHE con RSA	512 bits	RSA 3072 bits	4265,315	2558,398	40,02
DHE con RSA	1024 bits	RSA 3072 bits	4820,62	3211,268	33,38
ECDHE con RSA	163 bits	RSA 3072 bits	4532,768	2809,218	38,02
ECDHE con RSA	256 bits	RSA 3072 bits	4769,054	3103,64	34,92
ECDHE con ECDSA	163 bits	ECDSA 256 bits	1855,988	1205,437	35,05
ECDHE con ECDSA	256 bits	ECDSA 256 bits	2160,878	1505,485	30,33

Tabla 1. Resultados de las pruebas

Los resultados de las pruebas mostrados en la figura 3 reflejan una mejora en torno a un 50% por el uso de ECDSA para firmar y verificar los mensajes frente a una solución basada en RSA.

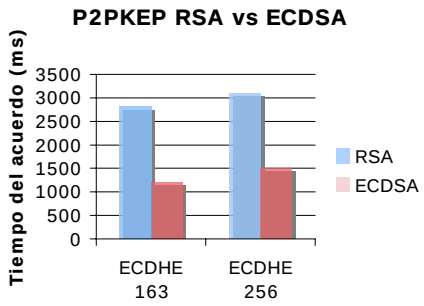


Fig. 3. Comparativa entre RSA y ECDSA.

En cuanto al algoritmo de intercambio de clave, los resultados de las pruebas (figura 4) muestran una mejora superior a un 12 % por el uso de ECDHE frente a DH.

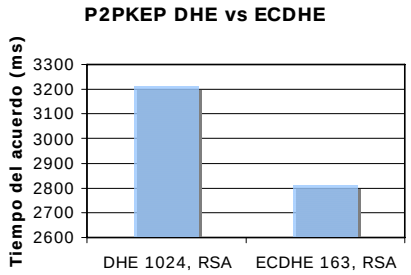


Fig. 4. Comparativa entre DHE y ECDHE con RSA.

V. Conclusiones

En este trabajo se presenta un protocolo, basado en el *Handshake de TLS*, para el intercambio de claves y la autenticación mutua en un escenario de redes ad-hoc. Para conseguir adaptar este protocolo a un entorno de redes ad-hoc, se han empleando mecanismos como el uso de UDP, la simplificación de las opciones criptográficas o su adaptación a un entorno de pares. Todo ello, junto con la utilización de criptografía de curvas elípticas, hace posible el empleo de este protocolo en entornos con dispositivos con pocos recursos como los que se han descrito. El material criptográfico negociado podrá utilizarse posteriormente para asegurar las comunicaciones a nivel de aplicación entre nodos participantes. Por tanto, la primera contribución del artículo sería la adaptación del *Handsake de TLS* para hacer viable su uso en este tipo de escenarios.

Como segunda contribución, se presenta una implementación inicial de este protocolo que en las pruebas realizadas evidencia una mejora del orden de 30-40% de P2PKEP respecto al *Handshake de TLS*. Las pruebas se han hecho con una red física y una implementación real de los procesos principales en ambos protocolos. Si bien la implementación no es completa, incluye los procesos más pesados, de forma que los resultados serían similares a los obtenidos en una implementación completa y por lo tanto las conclusiones son igualmente válidas.

Como ya apuntan otros autores,[12]) y mostramos en este trabajo, el uso de criptografía ECC en redes ad-hoc es altamente recomendable ya que proporcionan una mayor eficiencia de procesamiento conjunto para las operaciones de firma, cifrado y verificación. Además se ha verificado que el uso de certificados híbridos da un buen resultado en cuanto a tiempos de procesamiento.

Como conclusión, podemos afirmar que un escenario como una red VANET o una red de un escenario de emergencia, donde es razonable suponer la existencia de cierta infraestructura que provea a los nodos participantes con certificados, es viable, y en este artículo así se muestra, la utilización de un protocolo de autenticación mutua basado en certificados que, ofreciendo garantías de seguridad equivalentes o incluso superiores a TLS, es eficiente y lo suficientemente simple para ser ejecutado por nodos con pocos recursos.

VI. Trabajos futuros

En la propuesta actual se ha mantenido una estructura de mensajes y una secuencia de envío muy similar la que se emplea en el *Handsake de TLS*, lo cual nos ha permitido realizar pruebas comparativas de eficiencia. En actualidad, se está trabajando en la implementación completa del protocolo intentando mejorar aspectos como la flexibilidad y la eficiencia del mismo: aprovechando que en UDP los mensajes pueden llegar desordenados y/o perderse se ha diseñado un esquema de mensajes que permita enviar cada mensaje tan pronto como sea posible y pedir únicamente aquella información que estamos pendientes de recibir. También, se probará la robustez del protocolo ante ataques conocidos, pérdidas de paquetes, etc.

Agradecimientos

Este trabajo ha sido financiado por el proyecto T2C2 TIN2008-06739-C04-04.

Referencias

[1] J. Van Der Merwe, D. Dawoud and S. McDonald. “A Survey on Peer-to-Peer Key Management for Mobile Ad Hoc Networks”, *ACM Comput. Surv*. 39, 1, Article 1, Abril 2007

[2] Klaus Plobl, Thomas Nowey, Christian Mletzko, "Towards a Security Architecture for Vehicular Ad Hoc Networks,", pp.374-381, First International Conference on Availability, Reliability and Security (ARES'06), 2006

[3] E. de la Hoz et al., “PKI architecture for emergency and rescue MANET”, *IADIS internacional conference e-Society 2009*, pp. 201-204, Barcelona, Febrero 2009.

[4] L. Eschenauer, V. D. Gligor, & J. Baras. *On Trust Establishment in Mobile Ad-Hoc Networks*. Tech. Rep. MS 2002-10, Institute for Systems Research, University of Maryland, MD, USA, Octubre 2002.

[5] Gehrmann, C., Nyberg, K., Mitchell, C.J. The personal CA - PKI for a personal area network. In: IST Mobile and Wireless Telecommunications Summit, pp. 31–35, 2002

[6] T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol. Version 1.2*, RFC 5246, Agosto 2008.

[7] I. Riedel, *Security in Ad-hoc Networks: Protocols and Elliptic Curve Cryptography on an Embedded Platform*. Tesis de diplomatura. Ruhr-Universitat Bochum. Marzo 2003.

[8] E. Cronin, S. Jamin, T. Malkin and P. McDaniel, “On the Performance, Feasibility, and Use of Forward Secure Signatures”, *Proceedings of 10th ACM Conference on Computer and Communications Security* (CCS), pp. 131-144, Washington, DC, Octubre 2003.

[9] M. Raya, P. Papadimitratos, and J.-P. Hubaux, Securing Vehicular Communications, In IEEE Wireless Communications Magazine, Special Issue on Inter-Vehicular Communications, October 2006.

[10] IEEE Std 1609.2. “IEEE Trial-Use for Wireless Access in Vehicular Environment-Security Services for Applications and Management Messages”. 2006.

[11] J. Lee, G. Kim and S. Park, “Optimum UDP packet sizes in ad hoc networks”, *High Performance Switching and Routing*, 2002. Merging Optical and IP Technologies. Workshop on Volume, Issue, pp. 214-218, 2002.

[12] K. Lauter, “The Advantages Of Elliptic Curve Cryptography For Wireless Security”, *IEEE Wireless Communications*, pp. 62-67, Febrero 2004.

Implementación de un CAC basado en medidas de QoS para sistemas de Telefonía IP

José Mª Saldaña Medina, Julián Fernández-Navajas, José Ruiz Mas, Eduardo A. Viruete Navarro

Grupo de Tecnologías de las Comunicaciones (GTC) – Instituto de Investigación en Ingeniería de Aragón (I3A)
Universidad de Zaragoza
Ed. Ada Byron, C/ María de Luna nº 1, 50018 Zaragoza (España)
Tlf: (+34) 976 761 963 – Fax: (+34) 976 762 111
Email {jsaldana, navajas, jruiz, eviruete}@unizar.es

***Resumen*- Se presenta un sistema de CAC *(Call Admission Control)* basado en medidas de parámetros de Calidad de Servicio para Telefonía IP que utiliza SIP (*Session Initiation Protocol)*. En empresas con sucursales en varios países, permite que las llamadas internacionales se establezcan en dos tramos: uno a través de Internet con VoIP (*Voice over IP*), hasta un *gateway* VoIP-RTC (Red Telefónica Conmutada) en una sucursal del país destino, y otro por RTC hasta el usuario, con tarifa de llamada local. Las decisiones de CAC están basadas en medidas de Calidad de Servicio periódicas, en las tarifas de las llamadas, y en el número de líneas libres del *gateway*. El sistema se ha implementado en una plataforma de pruebas basada en virtualización.**

***Palabras Clave*- Telefonía IP, VoIP, CAC, MBAC, SIP, virtualización, QoS**

I. INTRODUCCIÓN

El uso de Internet para la realización de comunicaciones de voz en entornos corporativos permite reducir los costes de las llamadas. Las conferencias entre sucursales pueden realizarse utilizando VoIP (*Voice over IP*) (Fig. 1a). Para las empresas con presencia en varios países, una posible mejora es usar este sistema también para conferencias internacionales con destino a terminales tradicionales. Estas llamadas podrían llevarse a cabo mediante dos tramos, uno a través de Internet con VoIP hasta el país destino, y otro por RTC (Red Telefónica Conmutada) hasta el usuario, con tarifa de llamada local (Fig. 1b). De forma más concreta, una solución interesante para la empresa son los sistemas de Telefonía IP, que añaden a VoIP más servicios, disponibilidad y seguridad.

Los usuarios de estos sistemas buscan una Calidad de Servicio (*Quality of Service*, QoS) similar a la que proporciona la RTC. La VoIP es un servicio en tiempo real, en el que el retardo de los paquetes es uno de los parámetros que más afecta a la calidad de las llamadas. Un método para añadir QoS es el Control de Admisión de Llamadas (*Call Admission Control*, CAC) [1], que acepta o rechaza llamadas en función de los parámetros de QoS en cada momento.

Una mejora para el CAC es buscar la mejor ruta, en cuanto a parámetros de QoS y costes, para el establecimiento de las conexiones, teniendo en cuenta que pueden existir diversas ubicaciones disponibles desde las que establecer la llamada local.

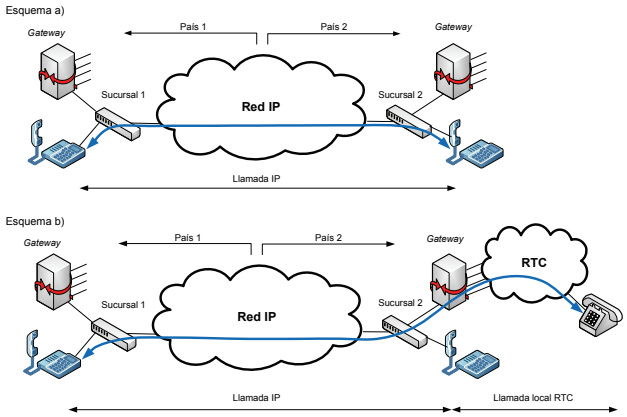


Fig. 1. Esquema tradicional y esquema propuesto

Para la implantación de un sistema de Telefonía IP a través de Internet, es conveniente implementarlo previamente en un entorno controlado. En el presente trabajo se propone el uso de virtualización, que permite desplegar un escenario de red completo dentro de una sola máquina física. Las ventajas de esta solución son que las aplicaciones probadas pueden ser implantadas sin apenas modificaciones, y que las implementaciones de los protocolos son reales.

El artículo está organizado de la siguiente forma: la sección II presenta la situación actual de los sistemas CAC para Telefonía IP. La sección III expone la arquitectura del sistema y su modo de funcionamiento. En la sección IV se describe la plataforma en la que se ha implementado el sistema. La sección V describe las herramientas *software* utilizadas. La última sección detalla las conclusiones de este trabajo.

II. PROBLEMÁTICA DE LA QOS: CAC

Para la señalización y el establecimiento de la llamada en el ámbito de la VoIP, se dispone de varias opciones como SIP (*Session Initiation Protocol*), H.323 o IAX (*Inter-Asterisk eXchange protocol*). El análisis desarrollado en este trabajo se ha centrado en SIP porque se trata de un protocolo sencillo y bien integrado en redes IP [2].

Al introducir un CAC en un sistema de Telefonía IP, se puede aceptar, rechazar o redirigir llamadas, según el estado de la red, consiguiendo así un control de QoS. El paradigma de aceptación de una nueva petición consiste en que, al aceptarla, las demás llamadas en curso no se vean afectadas viendo degradada su calidad, aumentando las pérdidas de paquetes y los retardos [3].

Uno de los tipos de CAC adecuados para la obtención de QoS es el CAC basado en medidas (*Measurement-based CAC*, MBAC) [4]. Actualmente estos sistemas son utilizados en algunas soluciones comerciales [5], pero están limitadas a los equipos del fabricante. En el caso de Cisco, por ejemplo, hay dos sistemas MBAC que funcionan para SIP: AVBO y PSTN Fallback [6]. Otros posibles sistemas [3] son los *Site-Utilization-Based* CAC (SU-CAC), y *Link-Utilization-Based* CAC (LU-CAC).

En todo caso, la implementación de un MBAC requiere el uso de herramientas de estimación y monitorización de parámetros de QoS. Existen herramientas que sirven para caracterizar diversos parámetros de una red: retardo, variación del retardo (*jitter*), ancho de banda máximo, ancho de banda disponible y tasa de pérdidas. Estas herramientas de medida pueden clasificarse en dos grandes grupos: extremo a extremo (*end-to-end*) y centralizadas. Las primeras se basan en la obtención de medidas desde los extremos de la red, sin preocuparse por su estructura interna. Por el contrario, las segundas utilizan información obtenida dentro de la propia red, como es la estadística de los *router*, para cuantificar los parámetros de QoS. En caso de no tener control sobre la red, las medidas a utilizar deben ser *end-to-end*. Otra posible clasificación divide las herramientas en activas [7] y pasivas [8].

III. Arquitectura del sistema

A. Descripción general del escenario

En el presente trabajo nos planteamos la implementación de un CAC basado en medidas activas *end-to-end* para un sistema de Telefonía IP. El esquema de partida es similar al de algunas soluciones propietarias, por ejemplo Cisco [3]. Este sistema se ha diseñado para un escenario de red que se corresponde con el de una empresa con sucursales en diferentes países (Fig. 2). Hay una PBX en un nodo central, y un agente local en cada sucursal. Se utiliza Internet como medio de transmisión del tráfico telefónico entre las sucursales, prescindiendo de otro tipo de enlaces.

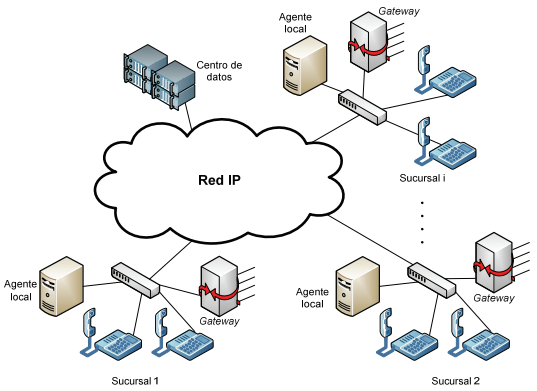


Fig. 2. Arquitectura del sistema.

El agente local es una máquina que introducimos en la red de cada oficina para gestionar la señalización SIP del sistema. Se encarga de que las llamadas sean cursadas por la mejor ruta teniendo en cuenta la QoS y los costes. Para ello, en primer lugar, unos procesos de estimación y monitorización de parámetros de QoS, ubicados en el agente local, realizan medidas entre sucursales, generando así la información necesaria para el sistema. Otro proceso implementa una función que, a partir de esos resultados, de las tarifas y del número de líneas libres y ocupadas del *gateway*, elabora unas tablas en las que se basarán las decisiones de CAC. Por último, el agente, a través de un *proxy* SIP que tiene incluido, procesa la señalización de las llamadas para implementar el mecanismo de CAC, basándose en información contenida en esas tablas.

Asumiremos las siguientes hipótesis, sobre las que estamos actualmente trabajando en nuestro grupo:

1) Se dispone de un sistema de medidas de QoS que se ajusta de forma continua a las características de la conexión.

2) Existe una función que calcula la decisión a tomar a partir de las medidas de QoS, las tarifas y la ocupación de las líneas de los *gateway*.

B. Funcionamiento del sistema de CAC propuesto

En el sistema existe una única PBX, que contiene el plan de numeración. En cada oficina existe un agente local, situado de manera que todos los mensajes de señalización entre los terminales y la PBX pasen a través de él. De esta forma puede introducir señalización para implementar las decisiones de CAC, y llevar cuenta de las llamadas establecidas en el *gateway* en cada momento.

Las llamadas internas a la sucursal son gestionadas por el agente local, y no necesitan acudir a la PBX para establecerse. En el caso de las llamadas entre sucursales, que no salen a RTC, las decisiones de CAC serían sólo de aceptación o no de la llamada, en función de la QoS, puesto que no tiene sentido redirigirlas a otra sucursal diferente a la del teléfono destino. Por último, si el usuario solicita una llamada a RTC, el sistema puede elegir el *gateway* a través del que se establecerá la llamada local. En muchos casos existirán varias opciones, por ejemplo si hay más de una sucursal en el país destino de la llamada, etc.

Para tomar las decisiones de CAC, cada agente local utiliza una "tabla de decisiones" (Tabla 1), en la que se especifica cómo actuar en el caso de recibir una petición de llamada (INVITE) desde una determinada sucursal.

Origen	Llamada interna	Llamada al *gateway* (RTC)
1	Aceptar / rechazar	Aceptar / rechazar / redirigir a i
2	Aceptar / rechazar	Aceptar / rechazar / redirigir a i
...	...	...
N	Aceptar / rechazar	Aceptar / rechazar / redirigir a i

Tabla 1. Tabla de decisiones

Esta tabla se construirá a partir de otras que veremos en el apartado siguiente, y que dependen de las estimaciones de QoS, de las líneas disponibles en cada *gateway*, y de la tarificación de cada sucursal.

Cuando el agente recibe un INVITE de la PBX (Fig.3), con destino a un usuario de su sucursal, lo acepta o rechaza en función de la entrada de la tabla que corresponda a la sucursal origen. La llamada puede ser rechazada bien por falta de QoS, o bien por no estar disponible el usuario destino.

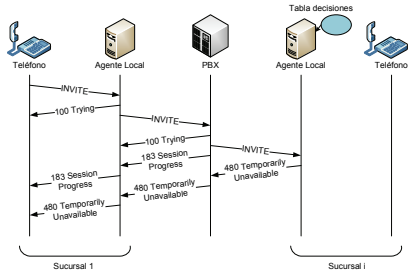


Fig. 3. Llamada entre sucursales rechazada

Si al agente local llega un INVITE de una llamada con destino al *gateway* y la tabla de decisiones indica *rechazar,* se enviará un mensaje SIP *"480 Temporarily Unavailable"* (Fig. 4), a la PBX. Ésta, según su plan de numeración, podrá acudir a otra sucursal que tenga tarifa económica para el destino de la llamada.

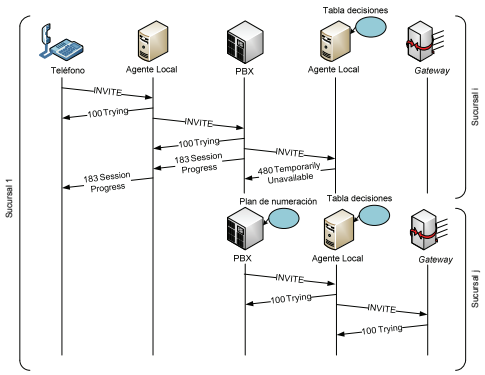


Fig. 4. La PBX intenta la llamada por el *gateway* de otra sucursal

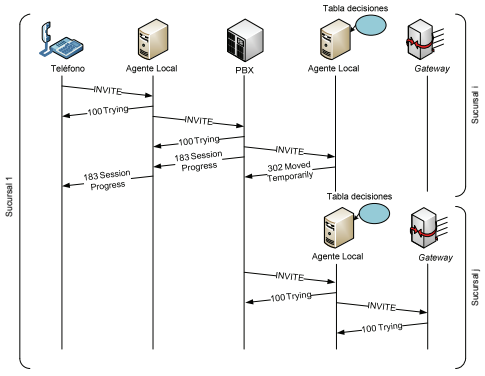


Fig. 5. Llamada redirigida de una sucursal a otra

Si la tabla de decisiones indica *redirigir*, el agente puede actuar también como *redirect server,* reencaminando la llamada a la sucursal por donde haya líneas libres para establecer la llamada (Fig. 5). El agente enviará un mensaje del tipo 3XX, informando sobre la ruta alternativa, y entonces la PBX intenta la llamada al nuevo destino, sin necesidad de acudir a su plan de numeración.

C. *Tablas de control*

Como hemos visto, la tabla de decisiones (Tabla 1) de cada agente se construye a partir de otras tablas en las que se refleja la QoS medida entre las distintas sucursales, la tarificación, así como las líneas totales y disponibles en cada *gateway*. Explicaremos ahora cada una:

1) Tabla de QoS (Tabla 2): En cada sucursal existirá una tabla con las estimaciones de cada parámetro de QoS. Los elementos de la diagonal no se definen, ya que representarían medidas dentro de una misma sucursal.

	Agente 1	Agente 2	...	Agente i
Agente 1	-	par 1→ 2	...	par 1→ i
Agente 2	par 2→1	-	...	par 2→ i
...	...	...	...	...
Agente i	par i→ 1	par i→ 2	...	-

Tabla 2. Tabla de QoS

2) Tabla de tarifas (Tabla 3): En cada campo se incluye el tipo de tarifa con varios niveles, representados por un número entero, correspondiente a una llamada desde el *gateway* de la red *i* hasta el país *j*.

	País 1	País 2	...	País j
Gateway 1	Tar 1→1	Tar 1→ 2	...	Tar 1→ j
Gateway 2	Tar 2→ 1	Tar 2→2	...	Tar 2→ j
...	...	...	...	...
Gateway i	Tar i→ 1	Tar i→ 2	...	Tar i→ j

Tabla 3. Tabla de tarifas

3) Tabla de líneas de cada *gateway* (Tabla 4): Permite conocer la disponibilidad de líneas con RTC disponibles en cada sucursal.

	Total Líneas	Líneas Ocup.
Gateway 1	TL 1	LO 1
Gateway 2	TL 2	LO 2
...	...	...
Gateway i	TL i	LO i

Tabla 4. Tabla de líneas

En el caso de que cada agente conociera sólo los parámetros de su sucursal en relación con las demás, la Tabla 2 sería solamente un vector columna. La tabla de líneas (Tabla 3) quedaría reducida en ese caso a un contador del número de líneas libres en el *gateway* de la propia sucursal. Por tanto, no se compartiría información con el resto de sucursales sobre las medidas de QoS (Tabla 2), ni sobre el número de líneas ocupadas en el *gateway* (Tabla 4). La tabla de tarifas sí se podría tener entera en cada agente, puesto que su periodo de actualización será muy largo, del orden de días, semanas, meses, etc.

IV. PLATAFORMA DE PRUEBAS

Una vez definida la arquitectura del sistema, pasamos a detallar la plataforma de pruebas en la que lo hemos implementado. Se ha buscado un diseño que se adaptase bien al sistema de Telefonía IP, y lograse emularlo con realismo, permitiendo pruebas y medidas con flexibilidad.

Para construir esta plataforma se podría recurrir a la simulación. Existen herramientas adecuadas, como OPNET o NS-2, pero no permiten el uso de las implementaciones concretas de los protocolos a utilizar en el entorno real.

También se podría construir la plataforma de pruebas con máquinas reales, pero supondría un coste muy elevado en equipos y elementos de red, dada la cantidad de dispositivos que integran el escenario.

Algunos estudios [9] han recurrido a la virtualización de varias máquinas en un único equipo físico, o en una pequeña red, para minimizar costes y optimizar el control del entorno de pruebas. Al virtualizar podemos disponer de un conjunto de máquinas, cada una con su sistema operativo, que se ejecutan sobre el *hardware* real de una sola máquina física. Es la solución que se ha elegido para el presente trabajo.

Dentro de los tipos de virtualización que se pueden distinguir, seleccionamos la solución de *paravirtualización* Xen, que permite una velocidad similar a la que se daría en un sistema no virtualizado [10].

La máquina utilizada tiene el Sistema Operativo CentOS 5. La versión del núcleo de Linux es la 2.6.18-8.1.15. Dispone de un procesador Core 2 Duo a 2.40 Ghz, 2MB de Cache nivel 2, y 4GB de RAM. Las máquinas virtuales tienen instalado también el Sistema Operativo CentOS 5. La versión de Xen instalada es la 3.03-25.0.4.

V. HERRAMIENTAS SOFTWARE UTILIZADAS EN LA IMPLEMENTACIÓN

Los elementos que componen el escenario propuesto (PBX, *softphone*, *proxy* SIP, *gateway* VoIP) deben tener poca carga computacional, dado que se van a utilizar en un entorno de máquinas virtuales. Se usarán soluciones de *software* libre.

Para la PBX utilizamos la versión 1.6. de Asterisk, de Digium, que se está utilizando en muchos entornos por su flexibilidad, actualizaciones y por su distribución bajo licencia GNU-GPL. Se ha utilizado un plan de numeración que permite redirigir la llamada a otra sucursal en el caso de que no sea aceptada por el *gateway* seleccionado como primera opción. En ese caso, la llamada se intentaría establecer a través del *gateway* de otra sucursal del mismo país.

Necesitamos también un *proxy* SIP que tenga opciones de *redirect server*, y que se pueda programar para implementar de este modo las decisiones de CAC. Debe ser capaz de consultar información externa, en una base de datos. La solución elegida ha sido OpenSIPS, continuación del proyecto OpenSER.

Utilizamos el *softphone* PJSUA, que tiene poca carga computacional, y funciona por interfaz de comandos. Forma parte del proyecto PJSIP, que ofrece bajo licencia GPL el *software* de una pila SIP completa.

Por último, necesitamos una manera para emular los *gateway* de un sistema de Telefonía IP. Hay que tener en cuenta que en nuestra plataforma no disponemos de conexiones reales con RTC. La solución adoptada ha sido el uso de PJSUA, ya que dispone de la posibilidad de establecer varias llamadas simultáneas, limitando ese número al de líneas del *gateway* a emular. De esta manera, cuando se encuentren ocupadas todas las líneas, se considera que ese *gateway* está totalmente ocupado, y rechazará las llamadas.

VI. CONCLUSIONES

Hemos definido la arquitectura de un sistema CAC para Telefonía IP, que está basado en medidas de QoS y utiliza el protocolo SIP. Permite que las llamadas internacionales se realicen en dos tramos: uno a través de Internet hasta el país destino, y otro desde un *gateway* en ese país, hasta el usuario final. Las llamadas son aceptadas, rechazadas o redirigidas a otra sucursal, según las medidas de QoS, las tarifas y el número de líneas libres de cada *gateway*.

Finalmente, el sistema se ha implementado en una plataforma de pruebas basada en virtualización, buscando para ello el *software* adecuado de PBX, *proxy* SIP, *softphone* y *gateway*.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado gracias al proyecto RUBENS (*Rethinking the Use of Broadband access for Experience-optimized Networks and Services*) del cluster europeo EUREKA CELTIC (código EU-3187 CP5-020), y al proyecto TSI-020400-2008-020 del subprograma AVANZA I+D del Ministerio de Industria, Turismo y Comercio.

Queremos por último agradecer la colaboración de los estudiantes de Proyecto Fin de Carrera Laura Esteban y Adrián Rejas, que han ayudado en la implementación del sistema y en la realización de medidas.

REFERENCIAS

[1] J. Yu, I. Al-Ajarmeh, "Call Admission Control and Traffic Engineering of VoIP", Second Intenational Conference on Digital Telecommunications (IEEE ICDT'07).

[2] P. Zave, "Understanding SIP through Model-Checking", Principles, Systems and Applications of IP Telecommunications. Services and Security Next Generation Networks: Second International Conference, IPTComm 2008, Heidelberg, Germany, jul 2008. pp 256 – 279.

[3] S. Wang, Z. Mai, D. Xuan, W. Zhao, "Design and implementation of QoS-provisioning system for voice over IP," Parallel and Distributed Systems, IEEE Transactions on , vol.17, no.3, pp. 276-288, Mar. 2006.

[4] Y. Jiang, P. J. Emstad, V. Nicola, and A. Nevin. "Measurement-based admission control: A revisit". In 17th Nordic Teletraffic Seminar, 2004.

[5] "SIP: Measurement-Based Call Admission Control for SIP", http://www.cisco.com/en/US/docs/ios/12_2t/12_2t15/ feature /guide/ftcacsip.pdf, última visita 3/2/2009.

[6] "VoIP Call Admission Control", http://www.cisco.com/en/US/docs/ios/ solutions_docs/voip_solutions/CAC.pdf, última visita 3/2/2009.

[7] I. M. Ivars and G. Karlsson, "PBAC: Probe-Based Admission Control," *Proc. QoFIS 2001*, 2001, pp. 97–109.

[8] C. Cetinkaya and E. Knightly, "Egress Admission Control", *Proc. IEEE INFOCOM 2000*, Marzo 2000.

[9] J. Zhou, Z. Ji, R. Bagrodia, "TWINE: A Hybrid Emulation Testbed for Wireless Networks and Applications". Proc. IEEE INFOCOM 2006.

[10] B. Quetier, V. Neri, F. Cappello, "Selecting A Virtualization System For Grid/P2P Large Scale Emulation", Proc. of the Workshop on Experimental Grid testbeds for the assessment of large-scale distributed applications and tools (EXPGRID'06), París, Francia.

GSIBot: Una Plataforma para el Desarrollo de Servicio de Bots

Miguel Coronado, Alejandro Marqués, Carlos A. Iglesias.
Depto Ingeniería de Sistemas Telemáticos
E.T.S.I.Telecomunicación Universidad Politécnica de Madrid
Ciudad Universitaria s/n 28040 Madrid.
miguelc@gsi.dit.upm.es, alejandrom@gsi.dit.upm.es, cif@gsi.dit.upm.es.

***Resumen*- El artículo presenta la plataforma GSI Bot para el desarrollo de servicios de bots. El objetivo de la plataforma es facilitar la asistencia en lenguaje natural a los usuarios que están usando un servicio, facilitando su integración multicanal (mensajería instantánea, Web, teléfono móvil) y el mantenimiento del servicio, con herramientas para la gestión, manejo y edición de la base de conocimiento de los bots. El sistema ha sido aplicado para un servicio de atención al cliente de una operadora, y está siendo aplicado a la gestión de dudas en e-learning.**

***Palabras Clave*-AIML, Bot, J2EE, Agente Inteligente.**

I. INTRODUCCIÓN

Prácticamente desde el comienzo de la andadura de los ordenadores, los programadores han tenido la ambición de crear programas que fueran capaces de mantener una conversación con un interlocutor humano. Posiblemente comenzaron como un juego o entretenimiento para el programador, pero hoy en día están muy extendidos y cada vez más perfeccionados. Los primeros bot conversacionales o chatterbots, como son conocidos, datan de la década de los 60, era necesario un ordenador dedicado a su funcionamiento y sólo una persona podía utilizarlos a la vez. En la actualidad, con el gran desarrollo de la Web es imposible imaginarlos desligados de Internet y son capaces de atender multitud de conversaciones simultáneamente. Ya no sólo aparecen como aplicaciones de demostración en la página del autor sino que se encuentran integrados, por ejemplo, en canales de IRC, como agentes en sistemas de mensajería instantánea, agentes expertos en FAQ, servicios de atención al cliente, sistemas domóticos...[1] Por supuesto forman parte de numerosas aplicaciones comerciales y su presencia es patente en áreas tan diversas como la enseñanza, el ocio o el marketing [2].

Sin embargo, aún queda mucho camino por recorrer: los bots conversacionales están lejos de ser totalmente inteligentes y sólo son capaces de hablar de aquello que se les ha enseñado, de lo que tienen conocimientos. Este comportamiento es semejante al de los seres humanos, con la diferencia de que la capacidad de aprendizaje del bot es limitada y que es totalmente pasiva. Enseñar a un bot conversacional (programar un bot conversacional) es un proceso largo, repetitivo, cuya complejidad aumenta conforme el conocimiento del bot se hace más basto y que requiere personal cualificado.

Este trabajo investiga el desarrollo de algoritmos y herramientas para facilitar el mantenimiento y creación de bases de conocimiento por parte de los usuarios finales. En particular, se ha trabajado en la definición de asistentes que permitan, mediante heurísticos, detectar conflictos cuando se amplía la base de conocimiento.

AIML [3], acrónimo de *Artificial Intelligence Markup Language* es la tecnología utilizada en el proyecto para programar las bases de conocimiento. AIML es un dialecto de XML específicamente diseñado para crear bots conversacionales. AIML es un lenguaje fácil de aprender que permite ampliar el conocimiento de un bot existente o crear uno desde cero en muy poco tiempo. Un conjunto de ficheros AIML forman, por tanto, la base de conocimiento de un bot conversacional. La especificación completa de AIML engloba un amplio conjunto de etiquetas [4]. La siguiente figura muestra un ejemplo de una regla AIML sencilla:

```
<category>
  <pattern>QUE TAL</pattern>
  <template>
    <think>
      <set name=”topic”>SALUDO</set>
    </think>
    Bien, ¿Y usted?
  </template>
<category>
```

Fig. 1. Ejemplo de código AIML.

AIML no necesita ser compilado sino que es interpretado por una plataforma para bots AIML. En el desarrollo del proyecto se ha utilizado ProgramD [5]. ProgramD es la plataforma de código abierto para bots AIML más utilizada en el mundo. Implementa todas las capacidades de AIML y soporta un número ilimitado de bots una simple instancia en el servidor [6].

AIML ofrece multitud de posibilidades a la hora de confeccionar patrones [7]. Ofrecen capacidades para mantener una conversación coherente (conversación causal) y para recordar datos relevantes del interlocutor tales como su nombre, edad e incluso aficiones (conversación personal). Los patrones en AIML son muy selectivos y potentes. Su escritura es sistemática y muy repetitiva, por lo que las bases de conocimiento escritas en AIML se prestan a su edición por parte de herramientas automatizadas. Sin embargo, es necesario hacer uso de algoritmos de análisis del código AIML para evitar que al añadir nuevos patrones AIML

interfieran en el conocimiento existente y se corrompan la base de conocimiento.

II. Gestión de Base de Conocimiento

Se pueden plantear muchas formas de automatizar la fase de enseñanza al bot. El sistema puede ser más o menos autónomo en función de la cantidad de decisiones que se toman sin consultar al usuario, decisiones de las que –por tanto- el sistema debe estar seguro –o bastante seguro. Un sistema totalmente autónomo sería capaz de, por ejemplo, leer un cierto número conversaciones guardadas hechas a través de clientes de mensajería instantánea e inducir a partir de ellas los patrones para generar AIML. Un sistema nada autónomo sería aquel que pidiera al usuario cada uno de los valores a introducir en cada uno de las etiquetas AIML. En tal caso el sistema se asemeja a un editor de XML.

A medida que la autonomía del sistema aumente también aumenta la probabilidad de cometer un error y por tanto, crear una entrada en la base de conocimiento que sea incorrecta: al aumentar la autonomía disminuye la fiabilidad. Si el sistema no es nada autónomo, y por tanto nada inteligente, todas las deducciones hechas se corroboran con el usuario y por lo tanto se le exige mayor nivel de cualificación en el uso de la herramienta.

Se pretende liberar de trabajo al usuario y desarrollar una interfaz intuitiva cuyo uso necesite la menor explicación posible. Para ello se ha alcanzado un compromiso entre autonomía y la fiabilidad desarrollando una interfaz semiautomática o asistida. El sistema toma aquellas decisiones sobre las que los algoritmos desarrollados poseen certeza absoluta. En el resto de casos presenta al usuario las decisiones tomadas y pide su corroboración. En estos casos se traducen las consideraciones propias de AIML a términos fácilmente entendibles por el usuario.

Interfaz de usuario

Se debe hacer un esfuerzo adicional en el diseño de la interfaz de usuario puesto que se pretende liberar totalmente al usuario de la necesidad de conocer el funcionamiento del sistema.

La aplicación desarrollada genera el código AIML necesario para responder lo que el usuario ha introducido como respuesta a un conjunto de frases que podríamos llamar sinónimas, que preguntan por lo mismo o hacen referencia a lo mismo.

La aplicación siempre solicita del usuario que escriba varias frases diferentes que signifiquen lo mismo (o parecido). Así se pretende inducir qué palabras deben aparecer en el patrón de código AIML. También se solicita al usuario que introduzca la respuesta que debe dar el bot cuando se recibe cada unas de estas frases como consulta.

El proceso de enseñanza está estructurado en 5 pasos. En el primero se solicita al usuario el conjunto de frases sinónimas, ya que es el punto de partida. En el segundo paso el sistema habrá analizado todas las frases para descubrir si hay palabras en plural o conjugaciones verbales y las sustituye por singulares e infinitivos. También se buscan sinónimos entre las frases introducidas por el usuario y se indica que se considerarán como la misma palabra. Ambas acciones se anuncian al usuario y se le da la opción de cambiarlas. En el tercer paso (cuya interfaz se muestra en la figura 2) las frases vuelven a ser analizadas para buscar las palabras clave, aquellas que deben incluirse en el patrón del código AIML. Los resultados se presentarán en forma de tabla coloreada en la que aparecen las frases introducidas por el usuario. Las palabras consideradas como palabras clave

Fig. 2: Extracto del paso 4 de la interfaz Web de Gestión de Preguntas.

marcadas en color. Una vez más se da la opción al usuario de modificar esta decisión y desestimar palabras que a priori se han considerado como palabras importantes u otorgar relevancia a otras. El cuarto paso pide al usuario las respuesta que el bot debe dar a cada una de las frases introducidas. También se solicita el valor de la URL a la que se hace referencia o que se quiere mostrar asociar a la respuesta en el caso de que exista alguna. El quinto paso es de comprobación. Entre el cuarto y quinto pasos el sistema genera el código AIML y lo incorpora a una copia de la base de conocimiento que denominaremos Base de conocimiento provisional. Se despliega un bot asociado a esta base de conocimiento para poder realizar un test corrección con el nuevo fichero de prueba para poder comprobar si existen interferencias de las nuevas reglas AIML introducidas con las existentes en la Base de conocimiento. En caso de que el resultado del test sea satisfactorio se dará la opción al usuario de conversar con el bot que incluye las reglas recién generadas. Si el bot ha aprendido la lección cumpliendo las expectativas del usuario se da la opción de guardar los cambios, lo que hará que se vuelquen a la base de conocimiento principal.

Arquitectura de gestión de Respuestas.

La arquitectura del sistema es más compleja que las descritas hasta el momento. Se centra en torno a la interfaz Web puesto que será el usuario quien irá autorizando las acciones a realizar. Esta formada por varios componentes.

El analizador morfológico, gestor de sustituciones y el analizador de palabras clave junto con la interfaz Web forman el componente gestor de preguntas. Este bloque es el que se encarga de generar el código AIML en función de las frases de entrada y las respuestas a las mismas introducidas por el usuario.

El Analizador Morfológico busca la raíz morfológica de cada una de las palabras presentes en las frases introducidas por el usuario en un diccionario morfológico. Sirve para reconocer singulares y plurales como la misma palabra así como tiempos verbales.

El Gestor de Sustituciones se encarga de analizar el AIML en busca de reglas que representen sustituciones. En el ámbito de este proyecto, se denominan sustituciones a las operaciones realizadas dentro del preprocesado que reciben las frases introducidas por el usuario y que transforman los plurales en singulares, los tiempos verbales en infinitivos y las palabras sinónimas entre sí. Este manejador guarda en memoria estas sustituciones para un fácil acceso y agrega las que el usuario haya incluido en la sesión de trabajo actual. El Gestor de sustituciones hace uso de un diccionario de sinónimos para proponer sustituciones de palabras por sus sinónimos favoreciendo así la comprensión del bot de mayor número de palabras.

El Analizador de Palabras Clave estudia en función de la posición en que aparecen las palabras clave en cada una de las frases introducidas por el usuario y de su frecuencia de aparición el código de AIML y el código para el Knowledge Test File que se debe generar. El diseño de este bloque es el más delicado puesto que debe tener en cuenta todas las características del lenguaje AIML y del idioma para el que se programa el bot a fin de que la base de conocimiento resultante sea escalable y a la vez funcione como desea el usuario.

Una vez generado el AIML y el código XML para el KTF el gestor de AIML y el gestor de KTF se encargan de modificar la base de conocimiento y el archivo KTF asociado a la misma respectivamente.

En la arquitectura se presenta una base de conocimiento provisional que será cargada por el servidor de bots en un Bot. Esta es la base de conocimiento que alberga los cambios hechos por el usuario y que permite al usuario y al módulo de prueba de Bots comprobar que los cambios introducidos en la misma funcionan como se desea y que no interfieren con el conocimiento previamente existente. Si ambas verificaciones concluyen satisfactoriamente la base de conocimiento provisional se copiará a la base de conocimiento principal.

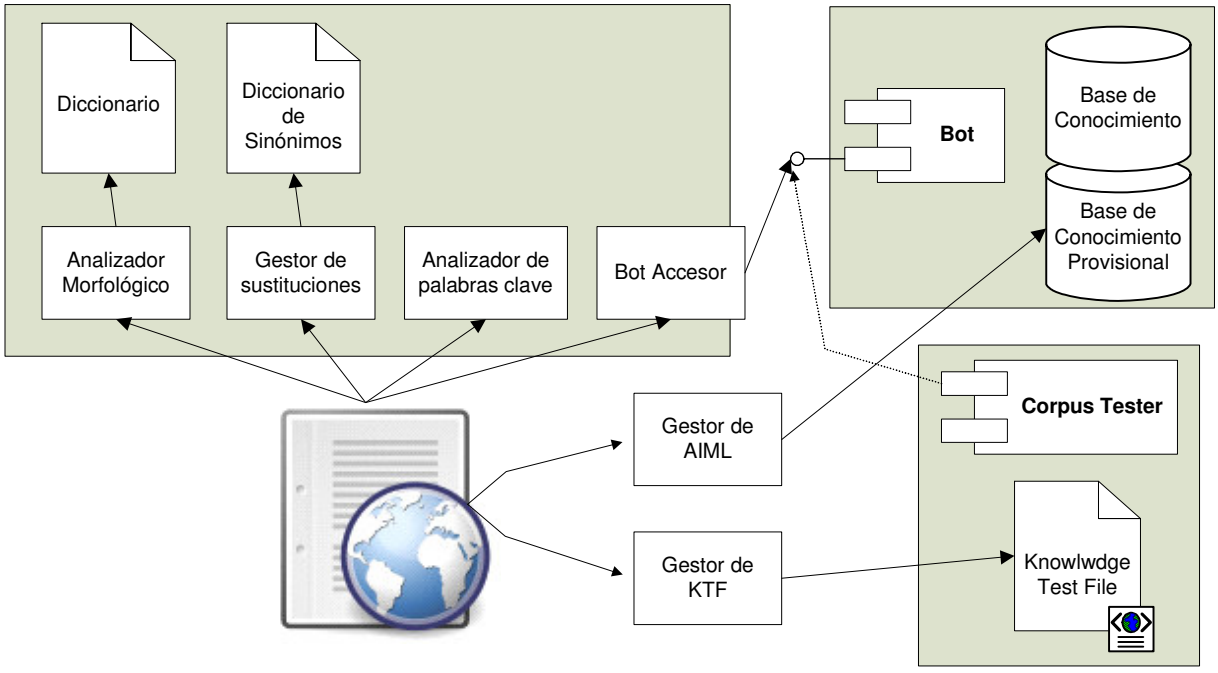


Fig. 3: Arquitectura de la herramienta Gestión de Preguntas.

Con esto se garantiza que, aunque en algún caso, el código AIML generado no cumpla con las expectativas, la base de conocimiento no se verá contaminada puesto que los cambios serán desechados.

III. Evaluación

Como conclusión se han desarrollado dos aplicaciones de edición de la base de conocimiento del bot: una para la edición de las respuestas ofrecidas por el bot y otra para añadir nuevo conocimiento de manera que el bot entienda un mayor número de frases de usuario descrita en este documento.

De forma paralela, para poder poner a prueba estos algoritmos y con el fin de generar estadísticas se ha desarrollado de un servidor de bots para el despliegue de bots. De tal manera que proporcionando los archivos que forman la base de conocimiento y los archivos de configuración se puede conversar con el bot por medio de peticiones HTTP con los parámetros adecuados. Esto facilita enormemente la creación de clientes de diversa índole.

Por último existe un módulo automatizado de pruebas de la base de conocimiento que a partir de unos ficheros de prueba comprueba si los cambios realizados en la base de conocimiento por los dos componentes anteriores interfieren con el conocimiento existente previamente.

El sistema desarrollado ha sido probado con éxito para la atención al cliente de una operadora, y actualmente se está investigando su integración para la teleeducación. En la figura siguiente se puede ver el interfaz del bot que se ofrece en la intranet de la operadora.

Fig. 4: Interfaz del bot desplegado en la intranet de una operadora.

IV. Conclusiones y trabajos futuros

En este trabajo se ha presentado una aplicación para gestionar, editar y mantener la base de conocimiento de un bot conversacional escrito en AIML de una manera eficiente, rápida, fácil y segura.

La línea de desarrollo futuro pasa por integrar todas estas aplicaciones en un framework así como proporcionar una interfaz Web conjunta para todas ellas.

Igualmente se planea desarrollar otras herramientas que permitan un control de versiones de la base de conocimiento de AIML, *merging* de bases de conocimiento entre las que existen interferencias, inclusión de los diccionarios de Open Office, módulo de estudio de logs de Bot para obtener estadísticas, creación de un servidor de bots mejorado (EnhancedProgramD)…

Agradecimientos

El proyecto ha sido realizado gracias a la colaboración con la Cátedra Orange a través de la financiación del proyecto "Aplicación de Técnicas Inteligentes al Desarrollo de Servicios de Valor Añadido basados en SIP".

Referencias

[1] Leonard, A. "Bots the origin of new species," New York: Penguin Books, 1997, pp. 152-153.

[2] Christine Frey "Digital 'buddies' are elaborate marketing tools, but their lifelike responses in online instant messages can be misleading." Los Angeles Times, Jul 18, 2001. Available: http://pqasb.pqarchiver.com/latimes/advancedsearch.html Los Angeles Times archive.

[3] A.L.I.C.E. AI Foundation Inc, "AIML – The Artificial Intelligence Markup Language", official website of the A.L.I.C.E Project. [Online]. Available: http://www.alicebot.org/aiml.html

[4] Richard Wallace and Noel Bush, "AIML 1.0.1 Specifications" [Online] Available: http://www.alicebot.org/TR/2001/WD-aiml/

[5] Noel Bush, "The home of ProgramD and xAIML technologies", official website of ProgramD Project. [Online]. Available: http://aitools.org/-Main_Page.

[6] Noel Bush and Kim Sullivan, "Getting Started with ProgramD" section :Configuration/Deployment. [Online] http://aitools.org/-Getting_Start-ed_with_Program_D# Configuration.2FDeployment

[7] Dr. Richard S. Wallace, "The Elements of AIML Style" 2004

Propuesta y evaluación de un esquema de caché para redes ad hoc

F.J. González-Cañete, E. Casilari, A. Triviño-Cabrera
Departamento de Tecnología Electrónica,
Universidad de Málaga
Campus de Teatinos, ETSI Telecomunicación, 29071, Málaga, España.
{fgc,ecasilari,atc}@uma.es

***Resumen*- En este artículo se describe y evalúa el rendimiento de un esquema de caché para redes ad hoc. En esta propuesta los nodos inalámbricos almacenan los documentos que solicitan en una caché local de forma que, sucesivas peticiones al mismo documento, pueden ser servidas directamente por su caché local en lugar de tener que acceder al servidor remoto. Por otro lado, los nodos pueden funcionar también como servidores para los demás nodos inalámbricos si interceptan las peticiones y sirven los documentos solicitados directamente usando su caché local. Finalmente, los nodos inalámbricos inspeccionan las peticiones y respuestas que retransmiten para aprender dónde y a qué distancia se encuentran los documentos. Usando esta información, los nodos inalámbricos pueden redireccionar las peticiones hacia otros nodos que tengan el documento y que se encuentran más cerca que el destino original de la petición. A través de simulaciones se evalúa el rendimiento de la propuesta planteada teniendo en cuenta el efecto del tiempo medio entre peticiones, el tiempo de expiración de los documentos, la distribución del tráfico y el tamaño de las cachés. Se demuestra que la propuesta reduce la latencia percibida por los nodos.**

***Palabras Clave*- Redes ad hoc, caché, política de reemplazo,**

I. INTRODUCCIÓN

Las redes MANET (*Mobile Ad Hoc Network*) ofrecen la oportunidad de ampliar la cobertura de los dispositivos inalámbricos de forma que nodos no conectados pueden comunicarse a través de la colaboración de dispositivos intermedios. Inicialmente, esta capacidad hace a las MANETs especialmente atractiva para casos de desastres o campos de batalla donde esas redes pueden funcionar sin infraestructura. De todas formas, el éxito de las comunicaciones inalámbricas se ha extendido al uso de MANETs en aplicaciones comerciales. En estos escenarios los usuarios requieren acceso a redes externas como Internet. Para esta conexión se necesita una pasarela (*Gateway*) que de acceso a Internet y a servidores externos. Desgraciadamente, la movilidad de los nodos en la red puede provocar que el *Gateway* esté temporalmente inaccesible. Las tecnologías Web deberían adaptarse a esta circunstancia para operar de forma adecuada. En este trabajo se estudia cómo el tráfico HTTP puede ser mejorado en una MANET si se usa cachés Web.

Cuando se usan cachés Web, los dispositivos almacenan en su caché los documentos que han sido previamente solicitados a un servidor HTTP. Los dispositivos móviles de la MANET pueden beneficiarse del espacio de almacenamiento de los otros nodos de forma que los documentos pueden ser servidos sin acceder al servidor HTTP. De esta forma, las peticiones HTTP pueden ser servidas incluso si el *Gateway* no está disponible.

Este artículo estudia cómo el uso de técnicas de caché parecidas a las usadas en la Web pueden funcionar en una red inalámbrica multisalto compuestas por nodos estáticos. El resto del artículo se encuentra estructurado como sigue: en la sección II se describe el esquema de caché, a sección III detalla el modelo de simulación y los resultados de las simulaciones y finalmente, la sección IV remarca las principales conclusiones y sugiere posibles trabajos futuros.

II. ESQUEMA DE CACHÉ

En esta sección se presenta un esquema de caché a nivel de aplicación para redes ad hoc. En este esquema los nodos de la red solicitan documentos que están situados en servidores de datos. Debido a las capacidades reducidas de los dispositivos inalámbricos, se supone que los servidores de datos no forman parte de la red, pero son accedidos a través de *Gateways* de Internet tal y como se especifica en [1]. El funcionamiento de la red sería como sigue: un nodo solicita un documento a un servidor de datos y la petición es enrutada a través de la red ad hoc usando el algoritmo de enrutado definido para la red. Cuando el servidor de datos recibe la petición responde enviando el documento al nodo. Este esquema cliente-servidor es muy similar al que se usa en Internet para el tráfico Web.

Tal y como ocurre en el tráfico HTTP, la forma más simple en que puede implementarse un esquema de caché es situando una caché local para cada usuario de la red, es decir, para cada nodo inalámbrico. Esta caché almacenará los documentos solicitados por cada nodo una vez que éste los recibe desde el servidor de datos. La siguiente ocasión en que el nodo necesite ese mismo documento, éste podrá ser servido directamente desde la caché local. A esta situación se le llama acierto en caché local y reduce drásticamente el tráfico en la red ad hoc, así como la energía consumida y el tiempo en recibir los documentos.

Como ocurre en los *proxies* HTTP los nodos de la red inalámbrica pueden adaptarse para interceptar las peticiones que retransmiten usando su caché local. A tal efecto, las funcionalidades del *proxy* son transferidas a los nodos de la red inalámbrica. Con esta tarea adicional, cada nodo en la ruta de una petición desde el nodo origen hasta el servidor de datos puede responder a la petición si tiene una copia válida del documento solicitado en su caché local.

La Fig. 1 muestra un ejemplo de red ad hoc, donde *DS* es un nodo servidor de datos, es decir, el nodo que almacena todos los documentos. A este nodo se accede a través de un *Gateway* (*GW*). Los nodos *1*, *2*, *3* y *4* son nodos de usuario que solicitan documentos a *DS*. Las conexiones entre los nodos indican los enlaces inalámbricos existentes.

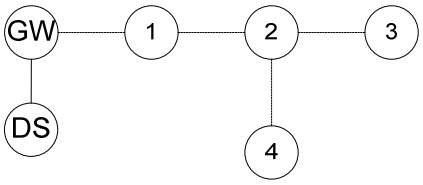


Fig. 1. Ejemplo de red ad hoc.

En el caso de que el nodo *2* solicite el documento *A*, la petición pasará a través del nodo *1* hasta *DS* usando el algoritmo de encaminamiento que tenga implementado la red. El servidor de datos responderá con el documento usando la ruta desde el nodo *1* al *2*. Finalmente, el nodo *2* almacenará el documento *A* en su caché local. Si el nodo *3* solicitara a continuación el mismo documento *A* a *DS*, la petición llegaría al nodo *2*, que comprobará si tiene una copia válida del documento *A* en su caché local y, si es así, respondería al nodo *3* con el documento. Esta intercepción de la petición reduce el número de saltos necesarios de seis (*3-2-1-DS-1-2-3*) en el caso de que no haya intercepción, a dos (*3-2-3*), con lo que la latencia que percibe el nodo *3* también se reduce.

Con el propósito de reducir la longitud de las rutas hacia el nodo que sirve las peticiones, se propone extender el esquema propuesto en el apartado anterior para tener en cuenta información acerca de la distribución de los documentos en la red. Más concretamente, se puede conseguir que los nodos almacenen información acerca de la distancia (medida en número de saltos) desde donde los documentos pueden ser encontrados. Esta información es extraída dinámicamente de los mensajes retransmitidos (tanto peticiones como respuestas). A partir de esos datos, cada nodo puede saber que un determinado documento está almacenado en la caché local de un nodo que está situado más cerca que el servidor de datos.

Para ilustrar este procedimiento, supongamos que el nodo *4* en Fig. 1 solicita el documento *A* a *DS*. La petición pasará a través de los nodos *2* y *1* hasta *DS*, de forma que el nodo *2* sabe que el nodo *4* tendrá el documento *A* y que se encuentra a un salto. Análogamente, el nodo *1* sabrá que el nodo *4* tendrá el documento *A* y que éste se encuentra a dos saltos. Cuando *DS* responda con el documento *A* a través de los nodos *1* y *2* hacia el nodo *4*, los nodos *1* y *2* anotarán también que *DS* tiene el documento *A* y que se encuentra a uno y dos saltos respectivamente. Si el nodo *3* solicitara posteriormente el documento *A* a *DS*, la petición llegaría al nodo *2*. En este momento el nodo *2* sabe que el nodo *4* y *DS* tienen el documento *A* y que se encuentran a uno y dos saltos respectivamente, de forma que el nodo *2* redireccionará la petición al nodo *4* ya que la ruta es más corta. Esta redirección reduce la cantidad de saltos de seis (*3-2-1-DS-1-2-3*) en el caso de que no haya redirección, a cuatro (*3-2-4-2-3*). Por lo tanto, la latencia percibida por el nodo 3 se reduce.

Desafortunadamente, la redirección tiene algunos problemas que deben tenerse en cuenta: la movilidad, la desconexión de los nodos y el reemplazo de los documentos en las cachés locales. Se debe implementar un sistema de alarma en el nodo que realiza las peticiones de forma que advierta esta situación y solicite el documento de nuevo ya que no va a poder ser servido tras la redirección de la petición. Con el fin de reducir la cantidad de errores de redirección causadas por el reemplazo de los documentos almacenados en las cachés locales se propone que cada nodo calcule la media de tiempo que los documentos permanecen almacenados en su caché local. De esta forma, cuando la información de redirección de un documento va a ser almacenada, este tiempo de expiración será el mínimo entre el TTL del documento y la media del tiempo que los documentos están almacenados en la caché local.

III. Modelo de simulación y evaluación

En esta sección se presenta el modelo de simulación utilizado así como la evaluación del rendimiento de una red ad hoc usando el esquema propuesto en el apartado anterior.

Las simulaciones se han realizado usando el simulador de redes NS-2.33 [2], que es el simulador más popularmente usado por los investigadores en redes ad hoc [3]. Se estudian tres escenarios en los que los nodos están uniformemente distribuidos formando una cuadrícula de tres densidades diferentes: 5x5, 7x7 y 9x9 nodos.

Existen 1000 documentos diferentes (identificados por un número) distribuidos entre dos servidores que están situados en esquinas opuestas (posiciones (0,0) y (1000, 1000)) del área de simulación. Para distribuir el tráfico, los documentos con un identificador par están situados en un servidor y los que tienen un identificador impar en el otro.

La Tabla 1 resume los principales parámetros de simulación.

Parámetro	**Por defecto**	**Valores**
Área de simulación	1000x1000	
Protocolo de encaminamiento	AODV [4]	
Número de nodos		5x5 – 7x7 – 9x9
Número de servidores	2	
Número de documentos	1000	
Número de peticiones por nodo	10000	
TTL (seg)	2000	500-1000-2000-4000-5000-Infinito
Tiempo medio entre peticiones (seg)	10	5-10-50-100
Pendiente Zipf [5]	0.8	0.4-0.6-0.8-1.0
Política de reemplazo [6]	LRU	
Tamaño de la caché (documentos)	100	25-50-100-200
Calentamiento (peticiones)	2000	

Tabla 1. Parámetros de simulación

Cada simulación se ha ejecutado cinco veces usando el mismo TTL para cada documento pero diferentes tiempos entre peticiones y orden de las mismas. La evaluación del rendimiento presentada es la media de los resultados obtenidos para las cinco simulaciones.

Se usan las siguientes métricas para cuantificar el rendimiento de la red: retardo (definido como el tiempo transcurrido entre la petición de un documento y la recepción de la respuesta) y la tasa de acierto en caché (definida como la proporción de documentos servidos por las cachés). En

cada nodo se distingue la tasa de acierto en caché local, de intercepción y de redirección como la proporción de documentos servidos por la caché local, por un nodo intermedio y por un nodo tras una redirección respectivamente.

Se compara el rendimiento de una red ad hoc en cuatro situaciones: 1) los nodos no implementan ningún tipo de mecanismo de caché (No caché), 2) los nodos únicamente tienen una caché local (CL – Caché local), 3) los nodos intermedios en las rutas hacia los servidores pueden interceptar las peticiones (I – Intercepción) y 4) los nodos implementan la redirección de las peticiones (Redirección) así como la intercepción.

Las figuras mostradas corresponden a la red de 5x5 nodos ya que los resultados obtenidos por las redes de 7x7 y 9x9 nodos son similares.

La Fig. 2 representa el retardo y la tasa de acierto en caché en función del tiempo medio entre peticiones.

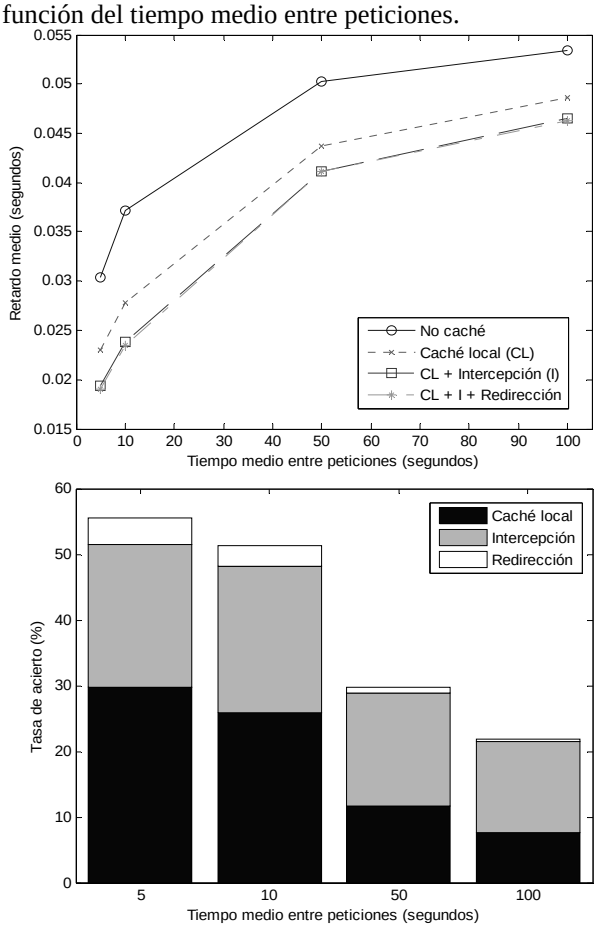


Fig. 2. Retardo y tasa de acierto en función del del tiempo entre peticiones.

El uso de la caché local reduce el retardo percibido por los nodos. Conforme el tiempo entre peticiones se incrementa, la media del retardo también se incrementa debido a la expiración de los documentos en las cachés locales. Este hecho causa la reducción de la tasa de acierto en caché local y, por tanto, aumenta la cantidad de documentos que tienen que volver a ser solicitados al servidor. La intercepción de peticiones mejora al uso exclusivo de la caché local y reduce aún más el retardo. Finalmente, la redirección no mejora el rendimiento de la intercepción debido a que la tasa de acierto es muy baja para todos los tiempos entre peticiones, e incluso llega a cero para un tiempo entre peticiones de 100 segundos. Para redes muy cargadas (una alta tasa de peticiones) la reducción del retardo en el caso de usar el esquema de caché propuesto es del 30% comparado con el esquema que no usa cachés. Esta reducción se mantiene constante incluso para redes poco cargadas (con una baja tasa de peticiones). En redes muy cargadas con nodos muy activos (tiempo entre peticiones de media 5) la cantidad de peticiones servidas por la caché local u otra caché intermedia ronda el 55%. Este hecho reduce drásticamente el tráfico en los servidores, distribuyendo la carga entre los nodos de la red.

La Fig. 3 muestra cómo el TTL de los documentos influye en el retardo y en la tasa de acierto.

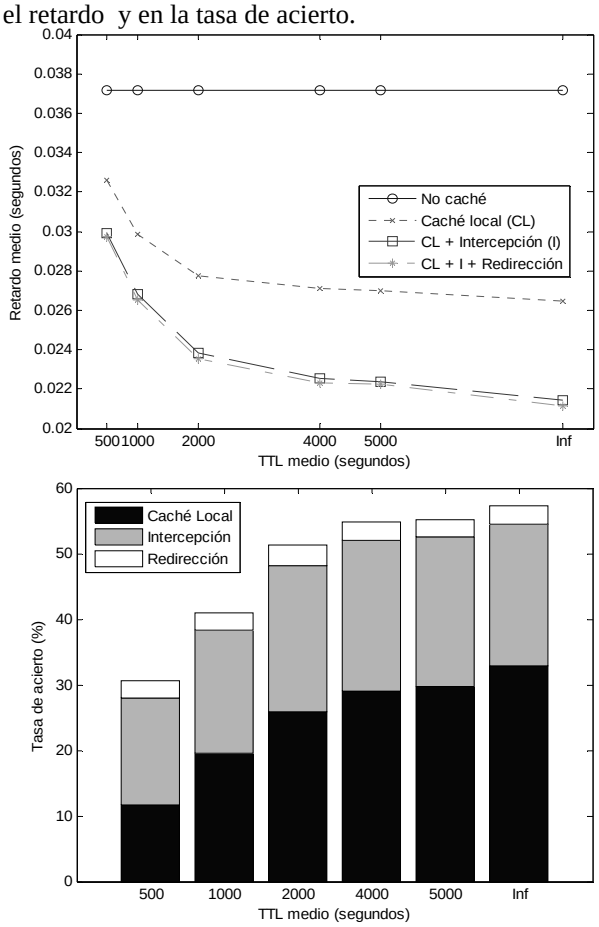


Fig. 3. Retardo y tasa de acierto en función del TTL de los documentos.

Conforme el TTL de los documentos se incrementa, el tiempo que pueden ser almacenados en las cachés también se incrementa, y por lo tanto pueden ser útiles durante más tiempo ya que expiran más tarde. Como puede observarse en la figura, el retardo se reduce conforme el TTL se incrementa. El retardo se reduce lentamente de forma asintótica hasta el valor óptimo en el que el TTL es infinito (el documento no caduca). En el caso de un TTL infinito el porcentaje de aciertos en alguna de las cachés ronda el 60%. Este hecho causa que la reducción del retardo sea del 40% comparado con el esquema sin cachés. En el caso de documentos que son modificados muy a menudo (bajo TTL) la reducción del retardo es del 20%. Las cachés locales y la intercepción de las peticiones claramente mejoran el esquema sin cachés. La redirección de las peticiones obtiene una baja y constante tasa de acierto para todos los valores del TTL y, por lo tanto, prácticamente no reduce el retardo.

La Fig. 4 compara el retardo y la tasa de acierto conforme cambia la pendiente de la distribución Zipf del patrón de peticiones.

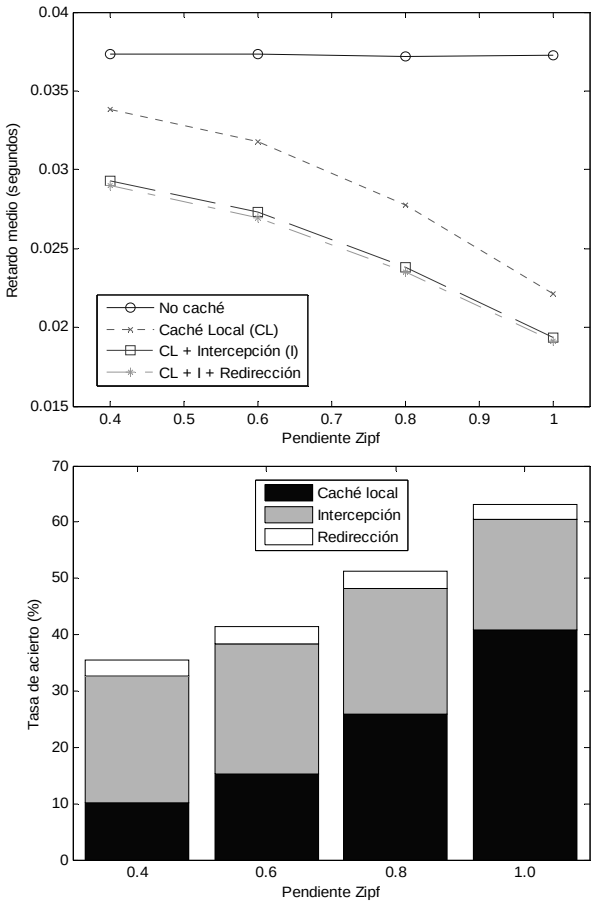


Fig. 4. Retardo y tasa de acierto en función de la pendiente Zipf.

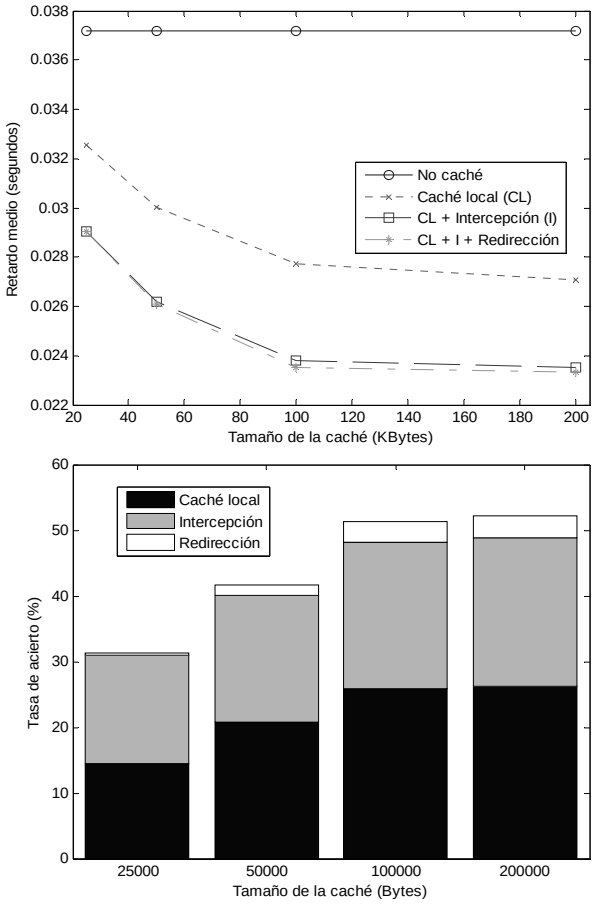


Fig. 5. Retardo y tasa de acierto en función del tamaño de las cachés.

Conforme la pendiente se incrementa, la tasa de aciertos en caché local también se incrementa debido al hecho de que los documentos más populares son solicitados más frecuentemente. Por otro lado, los aciertos de intercepción se decrementan conforme la pendiente Zipf se incrementa debido a que la mayoría del tráfico se sirve desde las cachés locales. Consecuentemente, el retardo se reduce conforme la pendiente Zipf se incrementa siguiendo un comportamiento similar al de los estudios anteriores. El retardo se reduce en un 20% y 40% para unas pendientes de 0.4 y 1.0 respectivamente.

Para terminar, la Fig. 5 muestra el retardo y tasa de acierto en cachés en función del tamaño de las mismas. Como en los estudios anteriores, la intercepción de las peticiones mejora al esquema sin cachés. La mejora en el rendimiento alcanza su límite cuando el tamaño de la caché es de 100 Kbytes (100 documentos), ya que los resultados obtenidos para una caché de 200 Kbytes (200 documentos) es similar. Para una caché de 25 documentos la mejora obtenida con la redirección de peticiones es casi cero, pero la tasa de acierto de redirección se incrementa conforme el tamaño de la caché se incrementa. Para una caché de 25 documentos, la reducción del retardo es del 20%, mientras que para cachés más grandes la reducción llega al 35%.

IV. CONCLUSIONES

En este artículo se ha propuesto un esquema de caché para redes ad hoc. Este esquema propone implementar una cache local en cada nodo de la red ad hoc para que cada nodo intermedio entre el nodo origen de la petición y el servidor pueda interceptar o redireccionar las peticiones. De esta forma la cantidad de saltos necesarios para completar la petición de un documento y su respuesta se reduce y, por lo tanto, también se reduce el retardo percibido por los nodos. Conforme el número de saltos se decrementa, la cantidad de mensajes que tienen que ser reenviados también se decrementa y, como consecuencia, el consumo energético se reduce. Se ha estudiado a través de simulaciones la influencia del tiempo medio entre peticiones (que define la tasa de peticiones), el efecto del TTL de los documentos, la influencia del patrón de tráfico y el tamaño de las cachés

Se puede concluir que el uso de las cachés locales combinado con el uso de la intercepción de los documentos reduce drásticamente el retardo que perciben los nodos. La redirección de las peticiones obtiene pobres tasas de acierto, por lo que la reducción que se produce en el retardo no es significativa si se compara con la intercepción de las peticiones.

REFERENCIAS

[1] Wakikawa, R., Malinen, J.T., Perkins, C.E., Nilsson, A., Tuominen, A.J.: Global Connectivity for IPv6 Mobile Ad Hoc Networks, draft-wakikawa-manet-globalv6-05.txt, Internet Draft, Internet Engineering Task Force, 2006.

[2] NS-2 Home page: http://isi.edu/nsnam/ns/

[3] Kurkowski, S., Camp, T., Colagrosso, M.: MANET Simulation Studies: The Incredibles. ACM's Mobile Computing and Communications Review, vol. 9, no. 4, pp. 50-61, 2005.

[4] Perkins, C. E., Belding-Royer, E. M., and Das, S.: Ad Hoc On Demand Distance Vector (AODV) Routing. IETF RFC 3561, 2003.

[5] Adamic, L.A., Huberman, B.A.: Zipf's law and the Internet. Glottometrics, vol. 3, pp. 143-150, 2002.

[6] Coffman, E.G., Dennings, E.J.: Operating Systems Theory. Prentice-Hall., 1973.

Utilización de Códigos Fountain para la Transmisión Fiable de Datos en Redes HomePlug AV

P.J. Piñero-Escuer, J.P. Muñoz-Gea, M.R. Liarte-López, J. Malgosa-Sanahuja, J. Vidal-Panalés
Departamento Tecnologías de la Información y las Comunicaciones,
Universidad Politécnica de Cartagena
Antiguo Cuartel de Antigones (Campus muralla de mar), 30202 Cartagena.
pedrop.escuer@upct.es, juanp.gea@upct.es, rosa.liarte@upct.es, josem.malgosa@upct.es, jesus.vidal@upct.es

***Resumen*—Los avances tenológicos están provocando que cada vez sea más necesaria la instalación de redes de comunicaciones en el hogar o en la pequeña y mediana empresa (PYME). La tecnología de red que más interes está despertando en este tipo de entornos es la PLC (Power Line Communications, concretamente el estándard HomePlug AV), que utiliza la infraestructura de cableado de baja tensión del edificio para el intercambio de información. A lo largo de este trabajo se estudiaron algunas características del canal PLC y a la vista de los resultados se presentan los códigos Fountain como un mecanismo eficaz para la transmisión fiable de datos en este tipo de redes, comparando las prestaciones ofrecidas por dichos códigos frente a TCP.**

***Palabras Clave*—Códigos Fountain, Códigos Online, Powerline Communications, Homeplug AV.**

I. Introducción

Cada vez es mayor la cantidad de dispositivos electrónicos ubicados en el hogar con facilidades de comunicación. Desde televisores y aparatos de radio interactivos hasta ordenadores, vídeo-consolas, cámaras digitales y teléfonos móviles, sin olvidar algunos de los equipos de la línea blanca (neveras, etc); todos ellos compartiendo dinámicamente el acceso a Internet propio del hogar. Se avecina, por tanto, una era en la que las comunicaciones dentro del hogar (in-home) van a tener un papel relevante dentro de la denominada Sociedad de la Información.

Existen varias alternativas en la actualidad que se podrían utilizar para desplegar una red in-home. Estas tecnologías se pueden dividir en tres categorías [8]:

- *Inalámbricas*
- *cableadas*
- *No-new-wires*

Las tecnologías *No-new-wires* son aquellas capaces de aprovechar las infraestructuras de cableado ya existentes en el edificio para el despliegue de la red. Dentro de esta categoría tenemos las tecnologías que utilizan la línea telefónica, el cable coaxial del operador de CATV o la red eléctrica para el intercambio de datos. Esta última es la que más interés está despertando actualmente entre la industria y la comunidad científica, ya que las otras dos presentan el inconveniente de que, al menos la mayoría de países Europeos, el número de puntos de conexión con la línea telefónica o con la red de cable es muy limitado.

El estándar más aceptado dentro de la tecnología PLC es Homeplug AV (HP audio-video, o simplemente HPAV). Este estándar fue desarrollado por la *HomePlug Powerline Alliance* [3] y proporciona un ancho de banda de hasta 150 Mbps sobre los cables de baja tensión existentes en cualquier edificio. Para la conexión de un dispositivo a una red de este tipo se utilizan los adaptadores HPAV, que disponen de un interfaz Ethernet para la conexión del dispositivo en cuestión y se conectan a la red eléctrica utilizando cualquier enchufe disponible en el hogar.

La mayoría del tráfico existente en la red privada convencional es un tráfico de datos y por tanto parece interesante realizar un estudio de cómo se comporta una red HPAV cuando se quieren realizar sobre ella transmisiones de este tipo. Este trabajo tiene como objetivo estudiar de manera detallada el comportamiento del canal de comunicaciones PLC y evaluar la posibilidad de utilizar codificadores de tipo *Fountain* para la transmisión fiable de datos a través de este tipo de redes.

La estructura general del artículo es la que se indica a continuación. En la sección II se realiza un análisis de los principales inconvenientes que presenta el canal de comunicaciones PLC para la transmisión fiable de datos. A continuación, basandonos en los datos anteriores, en la sección III se proponen los códigos Fountain (tambien denominados *Rateless*), concretamente los códigos Online, como mecanismo eficiente para la transmisión fiable de datos en redes HPAV. En la sección IV se realiza una evaluación de las prestaciones ofrecidas por los códigos Online en una red HPAV real. Finalmente, la sección V resume las conclusiones más destacadas de este trabajo de investigación.

II. Análisis del canal de comunicaciones PLC

En este apartado se va realizar un estudio detallado del comportamiento del canal de comunicaciones PLC. El objetivo es encontrar las características principales del mismo con el fin de implementar un mecanismo para la transmisión de datos lo más eficiente posible. Las características que se desean medir son:

- Capacidad (variable) del canal como consecuencia de la aparición de una fuente de ruido en la red eléctrica.
- Evaluación de posibles asimetrías en la red.

Para la realización de todas las medidas presentes en este artículo se han utilizado los dispositivos Homeplug AV PLE200 de la compañia Lynksys [4].

Tabla I
VELOCIDAD DE TRANSMISIÓN PARA DISTINTOS DISPOSITIVOS CONECTADOS A LA RED ELÉCTRICA. INTERVALOS DE CONFIANZA AL 95%

Dispositivo	Velocidad de transmisión [Mbps]
Sin Ruido	86.921 ± 0.131
Cargador tlf. móvil	60.600 ± 0.458
Ordenador portátil	84.045 ± 0.253
Batidora	82.236 ± 0.824
Calefactor	83.884 ± 0.142
Disco duro multimedia	86.379 ± 0.229
Monitor	76.303 ± 0.141
Estufa eléctrica	59.061 ± 0.780
Flexo	79.519 ± 0.118

A. Evaluación del modelo de capacidad variable del nivel físico

El primer objetivo será tratar de mostrar que los dispositivos HPAV son capaces de adaptar su velocidad de transmisión cuando se conecta a la red eléctrica una fuente de ruido. Como ya existen muchos artículos que detallan el comportamiento de la red PLC ante fuentes de ruido teóricas [10][11], en los experimentos realizados se han utilizado como fuentes de ruido aparatos eléctricos de uso frecuente en cualquier hogar (o empresa).

Para realizar las medidas se han utilizado dos ordenadores conectados a sus dispositivos HPAV mediante adaptadores Ethernet. Con objeto de eliminar el ruido introducido por los ordenadores, sus fuentes de alimentación se conectaron a una fase de la instalación eléctrica distinta a la fase en la que se realizaron las medidas. Debido a que en una instalación eléctrica convencional la única posibilidad de comunicación entre dos fases electricas distintas es a través de la estación transformadora (ET), se puede afirmar que la distancia entre dos elementos conectados a fases distintas será lo suficientemente grande para que no se produzcan interferencias entre ambos (en nuestro caso del orden de 300m). Tenemos entonces que la fase eléctrica donde se realizaron las medidas estaría inicialmente libre de ruido. Posteriormente se fueron conectando a la red electrica los elementos bajo estudio y se midió la reducción en la velocidad de transmisión producida. Los resultados obtenidos se muestran en la tabla I.

Se puede comprobar como los dispositivos que más afectan a la velocidad de transmisión son la estufa eléctrica y el cargador de teléfono móvil, seguidos del flexo y del monitor; mientras que los demás dispositivos no afectan significativamente a la misma. Observando la evolución temporal de la velocidad de transmisión cuando se conecta una fuente de ruido, se comprueba como en ausencia de ruido la velocidad de transmisión es superior a los 85Mbps, y cuando se conecta la fuente de ruido la velocidad se reduce aproximadamente a los valores indicados en la tabla I. Cuando se desconecta la fuente de ruido la velocidad de transmisión vuelve gradualmente a su valor inicial.

Vemos como las técnicas de modulación y codificación utilizadas por HPAV son capaces de adaptar la velocidad de transmisión del dispositivo en presencia de una fuente de ruido para evitar que se produzcan pérdidas de paquetes. Tenemos por tanto que el canal de comunicaciones PLC es un canal de capacidad variable en función del nivel de ruido en la red eléctrica.

B. Evaluación de las asimetrías de la red

La finalidad de los siguientes experimentos es demostrar que si una fuente de ruido provoca un descenso en las prestaciones de un módem, los demás módems de la red no tienen por qué verse afectados. Estas medidas se realizaron sobre el escenario formado por tres ordenadores, que denominaremos PC1,PC2 y PC3 respectivamente. Cada uno de los interfaces de red de los PCs se conecta a la red eléctrica utilizando un módem HPAV. De nuevo, para evitar interferencias, la fuente de alimentación de los PCs se conecta a una fase distinta de la red PLC.

En primer lugar se midieron las velocidades de transmisión entre los tres módems HPAV en un escenario sin ruido, obteniéndose medidas similares a las de los apartados anteriores. Posteriormente, tras conectar una fuente de ruido (cargador de teléfono móvil) junto al PC3 se volvieron a medir las velocidades de transmisión. La evolución de la velocidad de transmisión entre los equipos en función del tiempo se muestra en la figura 1. Se observa como la aparición de la fuente de ruido en el segundo 6 provoca un descenso muy importante en la velocidad de la conexión PC1-PC3 pero no afecta a la velocidad de la conexión PC1-PC2. Cuando la fuente de ruido desaparece en el segundo 12 la velocidad de transmisión se recupera y con el tiempo alcanzaría su valor inicial. Decir por último que las dos transmisiones mostradas en la figura no son simultáneas, aunque en ambos casos el patrón de ruido es exactamente el mismo.

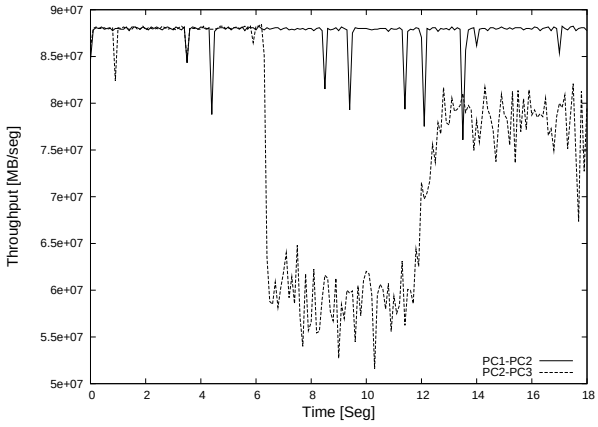


Fig. 1. Velocidad de transmisión desde un módem HPAV a otros dos módems, uno de ellos afectado por una fuente de ruido (PC1-PC2 y PC1-PC3 respectivamente)

Se ha comprobado como en las redes HPAV se pueden producir asimetrías muy importantes como consecuencia de la aparición de una fuente de ruido. Se observa como mientras algunos dispositivos funcionan de manera correcta, otros pueden ver reducida de manera muy importante su capacidad de transmisión. De todo ello se deduce que el paradigma *Overlay* (*peer-to-peer*) aplicado a las redes HPAV puede solucionar dichas asimetrías cambiando los roles de los *peers* en función del comportamiento de la red eléctrica.

III. CÓDIGOS FOUNTAIN

El protocolo más ámpliamente utilizado para la transmisión de datos en todo tipo de redes es TCP. Sin embargo, TCP está pensado para un canal de comunicaciones *full-duplex* y sus prestaciones bajan de una manera importante en canales *half-duplex* como es el caso del canal PLC. Una alternativa a TCP para la transmisión en este tipo de medios son los códigos *Fountain*. Estos códigos permiten la transmisión fiable de información sin necesidad de un canal de retorno. La naturaleza *half-duplex* de HPAV nos da a entender que utilizando este tipo de códigos para la trasmisión de datos se alcanzarán prestaciones similares o incluso mejores que con TCP.

Los códigos *Fountain* se basan en la idea de que el transmisor puede verse como una fuente de agua que es capaz de producir una cantidad infinita de gotas de agua. El receptor representa un recipiente que necesita recoger un cierto número de esas gotas para poder obtener la información. La ventaja principal que presentan este tipo de códigos es que el receptor puede recuperar la información sin importarle cuales de esas gotas ha recogido. Una codificación de tipo *Fountain* debe cumplir las siguientes características:

- El transmisor debe ser capaz de generar una cantidad potencialmente infinita de paquetes codificados a partir de la información que desea transmitir.
- El receptor debe poder decodificar un mensaje formado por K paquetes a partir de cualquier conjunto de K' paquetes codificados, para un valor de K' ligeramente superior a K.

Las tres implementaciones más importantes que existen en la actualidad de este tipo de códigos son los códigos LT [1], los códigos *Raptor*[6] y los códigos *Online*[2].

A. Códigos Online

Los códigos Online están definidos por dos parámetros, ϵ y q, y por el tamaño de bloque. Un mensaje de k símbolos de entrada, podría ser decodificado a partir de $(1 + 3\epsilon)k$ símbolos codificados con una probabilidad de error dada por la expresión $(\epsilon/2)^{q+1}$.

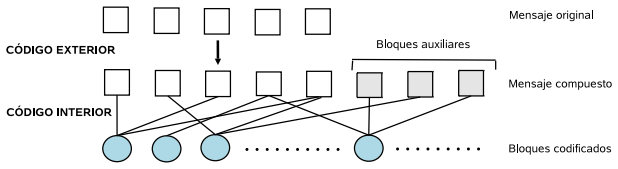


Fig. 2. Estructura de los códigos Online

La estructura general de esto códigos se muestra en la figura 2. Podemos ver como el proceso de codificación se divide en un código exterior y un código interior. El código interior se encarga de la generación de los bloques codificados, también llamados *check blocks*. Cada *check block* se calcula como la operación XOR de d bloques del mensaje a transmitir escogidos de manera uniformemente aleatoria (d representa el grado del *check block*). La probabilidad de que d=i viene dada por la siguiente distribución de probabilidad ρ_i:

$$\rho_1 = 1 - \frac{(1 + 1/F)}{(1 + \epsilon)} \tag{1}$$

$$\rho_i = \frac{(1 + \rho_1)F}{(F - 1)i(i - 1)} \quad i = 2, 3, ..., F \tag{2}$$

Donde el valor de F se obtiene de la siguiente forma:

$$F = \frac{ln(\epsilon^2/4)}{ln(1 - \epsilon/2)} \tag{3}$$

Debido a que la elección de los bloques del mensaje original es aleatoria, puede ocurrir que alguno de dichos bloques no se seleccione en la codificación. Para solucionar este problema se añade una codificación previa (código exterior) que genera $0.55qk\epsilon$ bloques auxiliares a partir del mensaje original. El conjunto de bloques del mensaje original más los bloques auxiliares se denomina mensaje compuesto (*Composite message*) y es la entrada del código interno. La redundancia introducida mediante el código externo permite que los bloques del mensaje original que no son seleccionados en el código interno puedan ser recuperados correctamente.

Para generar los bloques auxiliares se sigue el siguiente procedimiento: Para cada bloque del mensaje original se seleccionan q bloques auxiliares. Posteriormente, cada uno de los bloques auxiliares se calcula como la operación XOR de los bloques del mensaje original que se le han asignado.

Para poder realizar el proceso de decodificación el receptor debe conocer tanto el grado de cada bloque codificado como los bloques del mensaje compuesto por los que está formado (llamados bloques adyacentes). Se debe implementar por tanto algún mecanismo para enviar esta información al receptor. Una forma sencilla de solucionar este problema es que tanto el transmisor como el receptor utilicen el mismo generador pseudo aleatrorio para escoger los nodos del mensaje compuesto. De esta forma, basta con añadir una pequeña cabecera al bloque codificado que incluya el grado del mismo y la semilla utilizada para seleccionar los bloques adyacentes.

Una vez que el receptor posee toda la información necesaria, el proceso de decodificación que utiliza para obtener los bloques del mensaje compuesto a partir de los *check blocks* recibidos es el siguiente:

1) Encontrar un *check block* que solo tenga un bloque adyacente (d=1) y recuperar dicho bloque.
2) Eliminar el bloque recuperado de los demás *check blocks* de los que forma parte. Esto se puede conseguir sin más que aplicar de nuevo la operación XOR. Esto hace que el grado de los *check blocks* que contenían el bloque recuperado se decremente en una unidad y, por tanto, es muy probable que aparezcan nuevos bloques de grado uno.
3) Continuar con este proceso hasta recuperar una fracción 1-$\epsilon/2$ de los bloques del mensaje compuesto (suficiente para recuperar el mensaje original). Se puede observar claramente como el proceso de decodificación falla si en algún paso no existe ningún bloque de grado uno.

Una vez que hemos obtenido los bloques necesarios del mensaje compuesto, se pueden recuperar los bloques del mensaje original aplicando el mismo procedimiento.

IV. EVALUACIÓN DE LOS CÓDIGOS FOUNTAIN EN ENTORNOS PLC

En este apartado se evalúa la capacidad de los códigos Fountain para la transmisión fiable de datos en entornos HPAV.

Los resultados se comparan con el protocolo TCP, ya que es el más utilizado para la transmisión fiable de datos en todo tipo de redes de comunicaciones.

En este caso, las medidas se han realizado sobre un escenario con seis ordenadores conectados a la misma fase eléctrica que los dispositivos HPAV. Dos de dichos ordenadores realizan una transmisión continua de paquetes UDP a otros dos ordenadores del conjunto. Los dos ordenadores restantes actúan como transmisor y receptor de códigos Fountain y del protocolo TCP respectivamente. La distancia del cableado eléctrico entre estos dos últimos ordenadores es de unos 45m, que es de las mayores distancias que se pueden encontrar en una vivienda o local convencional (es decir, nos acercamos al peor caso posible). Con este escenario se consiguen dos objetivos: por un lado las fuentes de alimentación y los monitores de cada uno de los ordenadores introducían ruido a la red, y por otro, las transmisiones UDP simultáneas provocaban un descenso en la capacidad efectiva que el protocolo de acceso al medio (CSMA/CA) asignaba a cada equipo. Este descenso de capacidad producía pérdidas de paquetes como consecuencia del desbordamiento en los buffers de los módems HPAV.

La implementación de los códigos *Fountain* utilizada tiene como base la encontrada en [7]. A dicha implementación se le ha añadido la posibilidad de transmisión de los paquetes codificados en red mediante UDP. Para las transmisiones TCP se utilizó la aplicación SCP[5] que se basa en este protocolo para la transmisión de los datos. Las medidas realizadas consistieron en la transmisión de ficheros de tamaño comprendido entre 1 y 20 MB midiendo en cada caso el tiempo necesario para llevar a cabo la transmisión. Cada una de las transmisiones se repitió 5 veces y se obtuvo el intervalo de confianza al 95 % que se representa junto con los resultados en la figura 3

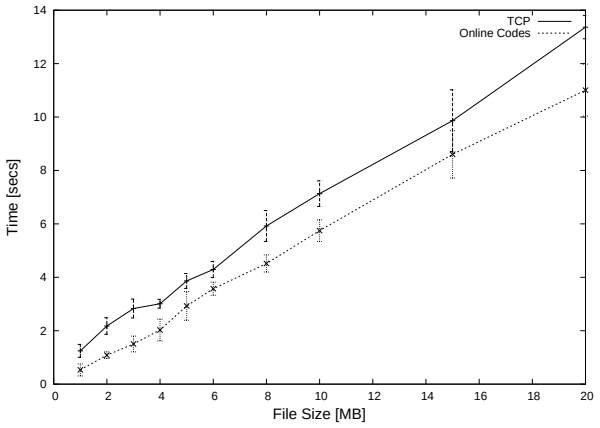


Fig. 3. Duración de transmisión de datos utilizando TCP y códigos Online sobre un entorno PLC con ruido y dos transmisiones simultáneas. Intervalos de confianza al 95%

Se puede observar como el tiempo necesario para transmitir el fichero con los códigos Online siempre es menor que el necesario para transmitirlo mediante la aplicación SCP. Esta diferencia de tiempos está en torno a 1 segundo para tamaños de fichero pequeños y aumenta hasta los 2 segundos cuando aumenta el tamaño del fichero. También se realizaron las pruebas con otras aplicaciones basadas en TCP como FTP pero no se muestran porque proporcionaban velocidades de transmisión inferiores a las obtenidas con SCP.

Mediante las medidas realizadas se ha comprobado que, en canales de acceso compartido, los códigos Online proporcionan unas prestaciones superiores que las aplicaciones basadas en TCP para la transmisión fiable de datos.

V. Conclusiones

A lo largo de este artículo se ha caracterizado de manera detallada el comportamiento del canal de comunicaciones PLC de baja tensión cuando se producen determinadas situaciones que afectan a la capacidad del mismo. Como consecuencia de las características observadas se pensó que los códigos Fountain reunían una serie de propiedades que podían ser de gran utilidad en este tipo de redes.

Observando los resultados obtenidos, se puede concluir que los codigos Fountain no solo son útiles para aplicaciones donde no existe canal de retorno o para transmisiones multicast, sino que también son una alternativa bastante interesante para la tranmisión de datos en canales de acceso compartido (p.e PLC). Mediante la utilización de estos códigos se consiguen tiempos de transmisión menores que con otros protocolos tradicionalmente utilizados para la tranmisión de datos como TCP. Los códigos *Fountain*, además, podrían facilitar la implementación de aplicaciones de tipo *peer-to-peer*, que podrían ser de gran ayuda a la hora de solucionar los problemas que provocan las asimetrías existentes en este tipo de redes.

Agradecimientos

Esta investigación ha sido apoyada por la subvención de proyecto TEC2007-67966-C03-01/TCM (CON-PARTE-1) y tambiém se ha desarrollado en el marco del “Programa de Ayudas a Grupos de Excelencia de la Región de Murcia”, de la Fundación Séneca, Agencia de Ciencia y Tecnología de la RM (Plan Regional de Ciencia y Tecnología 2007/2010). Pedro José Piñero Escuer también agradece a la Fundación Séneca la concesión de una beca predoctoral FPI.

Referencias

[1] M. Luby, *LT Codes*, Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on, pages 271–280.

[2] P. Maymounkov and D. Mazières. *Rateless codes and big downloads*. In Peer-to-Peer Systems II, Second International Workshop, IPTPS 2003, Berkeley, CA, USA, February 21-22,2003, Revised Papers, pages 247–255.

[3] HomePlug Powerline Alliance. *http://www.homeplug.org*

[4] Cisco-Linksys, 2009. *http://www.linksysbycisco.com/*.

[5] scp-secure copy, 2009, *http://www.mkssoftware.com/docs/man1/scp.1.asp*.

[6] A. Shokrollahi, *Raptor codes*. IEEE/ACM Trans. Netw., 14(SI):2551–2567, 2006.

[7] Implementation of Online Codes, 2009, *http://sourceforge.net/projects/onlinecodes/*.

[8] Y.-J. Lin, H. A. Latchman, R. E. Newman and S. Katar. *A comparative performance study of wireless and power line networks*. IEEE Communications Magazine, 41(4):54–63, 2003.

[9] D. J. C. MacKay. *Fountain Codes*. IEE Proc. Commun., vol. 152, no. 6, Dec. 2005.

[10] B. Jensen. *Experimental Studies of the Noise Recovery Ability of In-House Powerline Equipment*. IEEE Proceedings of ISPLC2008, pp. 269-273, 2008.

[11] B. Jensen, S. Kjaersgaard *Benchmarking and QoS of In-House Powerline Equipment under Noisy Conditions*. IEEE Proceedings of ISPLC2007, pp. 17-22, 2007.

SISTEMA DE INYECCIÓN Y MONITORIZACIÓN DE TRÁFICO SINTÉTICO EN SEGMENTOS DE ALTA CAPACIDAD

Alberto Pineda Rodríguez, Armando Ferro Vázquez, Alejandro Muñoz Mateos
Departamento de Electrónica y Telecomunicaciones
Universidad del País Vasco / Euskal Herriko Unibertsitatea
ETSI, Alameda de Urquijo s/n 48013 Bilbao
apineda001@ikasle.ehu.es, armando.ferro@ehu.es, alex.munoz@ehu.es

***Resumen*- Con el aumento de la velocidad en las redes de datos la capacidad computacional de la infraestructura de comunicaciones y de los servidores principales puede verse comprometida. Para poder probar el rendimiento de las redes, de los equipos de interconexión y de los servidores se hace necesario disponer de sistemas de inyección de tráfico, que permitan la generación de tráfico sintético con diferentes características. También es necesario disponer de sistemas de monitorización, que permitan estudiar las características del tráfico. Hacer esto en segmentos de alta capacidad de una forma eficaz no es un asunto trivial. Este artículo propone el diseño de una arquitectura de inyección y monitorización de tráfico sintético que pretende mejorar los rendimientos de las soluciones disponibles utilizando una arquitectura de propósito general como es un sistema Linux sobre un PC con una interfaz de red común. El fundamento en la mejora de los rendimientos se basa principalmente en la introducción de la lógica de inyección y monitorización en el núcleo del sistema operativo.**

***Palabras Clave*- Inyector de tráfico, monitorización de red, DAG, kernel, subsistema de red, análisis de tráfico, modelos de tráfico, timestamps.**

I. INTRODUCCIÓN

Cada vez es más común disponer de redes locales con anchos de banda importantes. Así, hablar de velocidades de 1 Gbps es algo habitual. La limitación fundamental en la evolución de esas redes no viene marcada por las líneas de comunicación sino por los equipos de interconexión. También hay que tener en cuenta la capacidad de nuestros equipos a la hora de procesar la información recibida de dichas redes ya que la escalada de la capacidad computacional no está siendo tan acusada. Por todo esto, muchos de los problemas que se dan en estas redes se deben a que los equipos de interconexión e, incluso, los propios servidores son incapaces de atender un tráfico tan elevado por un dimensionamiento inadecuado de su capacidad computacional.

Para poder avanzar en el estudio de esta problemática hay que disponer de herramientas adecuadas. Se necesitan sistemas de inyección de tráfico que permitan probar el producto en condiciones de estrés. Por otro lado, también se precisan herramientas de monitorización de tráfico que faciliten el estudio del comportamiento del tráfico dentro de la red y dentro de la propia arquitectura del sistema en estudio.

En este trabajo se propone el diseño de un sistema de inyección y de otro de monitorización basados en la utilización de una arquitectura de propósito general. Esta propuesta buscar mejorar el rendimiento trasladando la lógica al núcleo del sistema, más cerca del hardware.

Se presentan por separado los diseños del sistema de inyección y del de monitorización. Además se pretende que ambos sistemas sean tan genéricos como sea posible.

Para el caso de inyección, los parámetros principales que hay que controlar son la tasa de inyección y el modelo de tráfico sintético que hay que generar. Se pretende llegar a tasas que saturen enlaces Gigabit Ethernet siguiendo una distribución de tráfico determinada.

Por otro lado, las aplicaciones del sistema de monitorización son variadas: medir retrasos en redes, estudio de distribuciones de tráfico, estudio de la modificación de la distribución de tráfico por algún sistema intermedio, etc. En el caso que ocupa este artículo, se precia monitorizar el tráfico, entre otras cosas, para medir la bondad del inyector propuesto.

El tema se desarrolla siguiendo un esquema dividido en secciones. En primer lugar se presentan las soluciones tecnológicas disponibles separadas en dos apartados. Primero, los posibles sistemas de inyección y, segundo, los sistemas de monitorización. El siguiente apartado trata las distribuciones de tráfico que resultan de interés para la generación de tráfico sintético para este caso. A continuación se expone la propuesta de diseño donde se explica la solución en líneas generales. En los siguientes dos apartados se presentan las arquitecturas de referencia tanto para la monitorización como para la inyección. Por último se proponen en un apartado los casos de aplicación y la experimentación. Primero se describen los casos de aplicación de referencia. En segundo lugar se presenta el escenario de pruebas.

II. SISTEMAS DE INYECCIÓN

Existen varias posibilidades que dan solución a esta necesidad. Así se pueden encontrar inyectores software que permiten la generación e inyección de tráfico. Para este caso hay un gran número de aplicaciones gratuitas como: D-ITG, KUTE, pktgen, etcétera [1] [2] [3] [4] [5].

También se puede recurrir a soluciones hardware basadas en diseños para FPGA o a tarjetas específicas de captura e inyección de tráfico como las DAG que ofrece el spin-off Endace [6].

A. Inyectores software

Las opciones contempladas en este apartado hacen uso de la tarjeta de red de propósito general que incorpora el equipo y, por tanto, del subsistema de red de Linux.

1. Distributed Internet Traffic Generator (D-ITG)

Se trata de una plataforma que permita la inyección de tráfico reproduciendo procesos estocásticos en las variables aleatorias que definen el tiempo entre paquetes y el tamaño de los mismos [1] [2]. Tiene una arquitectura multi-componente formada por: ITGSend, ITGRecv, ITGManager e ITGLog. La comunicación entre el emisor y el receptor se efectúa por un canal de señalización independiente utilizando el protocolo TSP.

La plataforma de inyección D-ITG, en entornos Gigabit Ethernet, alcanza tasas de 612 Mbps [2].

2. Kernel-based UDP Traffic Engine (KUTE)

Esta aplicación se ejecuta a nivel de kernel. En este caso, en el kernel Linux 2.6. Con la ejecución a nivel de kernel se trata de mejorar el rendimiento de la aplicación [2].

Divide su funcionamiento en dos componentes: KUTE sender y KUTE receiver. Este inyector está pensado para inyectar tráfico a altas tasas y realizar medidas de throughput y jitter.

B. Inyectores hardware

Para la inyección de tráfico también se puede recurrir a soluciones hardware. En este caso, se contemplan tres opciones: tarjetas DAG de Endace, un diseño basado en FPGA y los sistemas embebidos que proporciona Gateworks.

1. Tarjetas DAG

Las tarjetas DAG ofrecen un interfaz para la inyección de tráfico. Sólo operan en el nivel 2 y no depende de la gestión del subsistema de red [6].

Endace proporciona, junto con la tarjeta, una serie de aplicaciones que permiten la generación e inyección de tráfico a través de la misma.

2. Diseño basado en FPGA

Aquí se contempla la posibilidad de realizar un sistema de propósito específico cuya función sea la de generar e inyectar tráfico.

Para realizar este sistema se requiere una tarjeta con interfaz Ethernet de 1 Gbps, con una FPGA y con interfaz de programación desde PC. Este tipo de tarjetas las proporciona, entre otros, Avnet.

3. Sistemas embebidos Gateworks con Network Processor

Gateworks ofrece dos familias de sistemas embebidos que están especialmente diseñados para aplicaciones de red.

Estos sistemas están compuestos por uno o varios puertos Ethernet 10/100, interfaz serie USB ó RS-232, bus Mini-PCI y un network processor de la serie Intel Xscale.

Además, estas tarjetas incorporan distribuciones Linux para sistemas embebidos (Open WRT o µLinux dependiendo de la familia).

III. SISTEMAS DE MONITORIZACIÓN

Para implementar el sistema de monitorización se puede recurrir, nuevamente, a soluciones software y hardware.

A. Monitorización software: libpcap

Dentro de las posibles soluciones de monitorización software se hace especial hincapié en la librería libpcap. Ésta proporciona un interface de alto nivel para la captura de paquetes [7].

Esta librería proporciona al programador una serie de funciones que permiten crear una aplicación que capture el tráfico que pasa por la tarjeta de red del equipo. Permite hacer marcas de tiempo que dependen del reloj del kernel de Linux.

B. Monitorización hardware

Dentro de este apartado se presentan soluciones ya propuestas en el trabajo anterior [8].

1. Tarjetas DAG

Estas tarjetas proporcionan también un interfaz de captura de tráfico. Además Endace incorpora una aplicación que gestiona dicha captura.

Las tarjetas DAG realizan un marcado de tiempo a nivel hardware que facilita el seguimiento de la distribución del tráfico en la línea de comunicación.

2. Sistema embebidos Gateworks con Network Processor

En este caso, asimismo, se contempla la opción de hacer uso de un sistema embebido de Gateworks para la monitorización de tráfico.

IV. LIMITACIONES DE LAS SOLUCIONES TECNOLÓGICAS

Las soluciones tecnológicas planteadas en el estado del arte presentan una serie de limitaciones que no las hacen del todo válidas para das solución al sistema objeto de este artículo.

A. Limitaciones en los sistemas de inyección

La principal limitación de los sistemas de inyección software planteados es que la tasa de inyección es insuficiente para saturar enlaces Gigabit Ethernet. Además, en el caso de la aplicación D-ITG el control se realiza desde plano de usuario. Esto genera problemas de rendimiento

asociados a consumos computacionales indeseados y a pérdidas de control del tiempo debido a la política de planificación.

Pasando a los inyectores hardware, la tarjeta DAG presenta varios problemas. Primeramente, la aplicación que permite la inyección de tráfico con dicha tarjeta no permite que el flujo inyectado siga ninguna distribución específica. En segundo lugar, dicha aplicación se ejecuta en espacio de usuario lo cual hace que aparezcan los problemas de rendimiento descritos para D-ITG. Por último, no se trata de una solución genérica.

También se ha propuesto la realización de un diseño basado en FPGA. Esto requiere un amplio conocimiento no sólo de electrónica sino también de las herramientas de diseño que se utilizan.

Por último, se ha presentado los sistemas embebidos que ofrece Gateworks. Las principales limitaciones de estos sistemas se centran en la escasa memoria y capacidad computacional. Además los interfaces Ethernet son 10/100, lo cual proporciona tasas de inyección bajas para esta aplicación.

B. Limitaciones en los sistemas de monitorización

Dentro de la monitorización software, las principales limitaciones que presenta la librería libpcap se deben a su ejecución en espacio de usuario. Estas limitaciones ya han sido comentadas.

La utilización de la DAG hace la solución poco genérica. Por último se han propuesto los sistemas Gateworks. Estos sistemas presentan escasa capacidad computacional y poca memoria para realizar labores de monitorización.

V. GENERACIÓN DE TRÁFICO SINTÉTICO

En este punto se presentan las dos distribuciones de tráfico que se ha considerado fundamentales [9].

A pesar de que existen técnicas que permiten un modelado más real del tráfico, como MMPP, se han escogido dos distribuciones básicas ya que el principal objetivo de este trabajo es obtener un sistema con un buen control del tiempo.

A. Distribución exponencial o de Poisson

Esta distribución se ha utilizado tradicionalmente para modelar redes telefónicas.

La ecuación (1) presenta la función densidad de probabilidad de la distribución.

$$f(x) = \begin{cases} \lambda \cdot e^{-\lambda \cdot x} & x \geq 0 \\ 0 \quad \textit{en otro caso} \end{cases} \tag{1}$$

B. Distribución Pareto

Es una distribución de probabilidad continua con dos parámetros a y b.

La ecuación (2) presenta la función densidad de probabilidad esta distribución.

$$f(x) = \frac{a \cdot b^{a}}{x^{a+1}} \tag{2}$$

VI. ARQUITECTURA DE REFERENCIA PARA EL INYECTOR

Se propone un inyector basado en el subsistema de red de Linux [10]. La solución que se plantea tiene dos objetivos fundamentales, por un lado, se pretende que la solución sea lo más genérica posible y, por otro, se desea tener un buen control del tiempo para poder generar distribuciones de tráfico fieles.

Para conseguir una solución genérica se ha decidido que se puedan utilizar dos interfaces de inyección de tráfico. O bien, una tarjeta de red genérica, o bien, una tarjeta DAG. Cualquier máquina puede disponer de una tarjeta de red genérica de una manera sencilla. Sin embargo, es interesante incluir la DAG en la solución ya que permite la inyección de tráfico a altas tasas (tasas de saturación del enlace Gigabit Ethernet). Como se plantean dos interfaces de inyección de tráfico, hay que establecer un interfaz común de dichas tarjetas con la aplicación. Dicho interfaz es el subsistema de red de Linux. Para poder llevar a cabo este diseño hay que modificar el módulo de kernel que hace las veces de driver de la DAG ya que estas tarjetas no hacen uso del subsistema de red para inyectar tráfico.

Por otra parte, para conseguir un buen control del tiempo se implementa la lógica de la aplicación a nivel de kernel. De esta forma se evitan los cambios de contexto indeseados, las recopias innecesarias, etc. obteniendo, de esta manera, unas distribuciones de tráfico más fieles debido a un mejor control del tiempo. En definitiva, se obtiene un mejor rendimiento.

En la figura 1 se plantea la arquitectura de referencia del inyector. Dicha arquitectura, como se observa en la figura, está compuesta por los módulos de kernel: trans_traffic, gen_traffic, gen_distrib y temp. El primer módulo, trans_traffic, se encarga de recoger los datos de configuración y de enviárselos a gen_traffic y gen_distrib. Gen_traffic se encarga de crear el flujo de tráfico sintético y almacenarlo en un lugar conocido. Gen_distrib, por su parte, se encarga de mandar cada paquete al subsistema de red para que sea transmitido en función de la distribución que se quiera seguir. El módulo temp se encarga de contar el tiempo que se le indique. Este módulo será utilizado por gen_distrib para contar el tiempo entre paquetes.

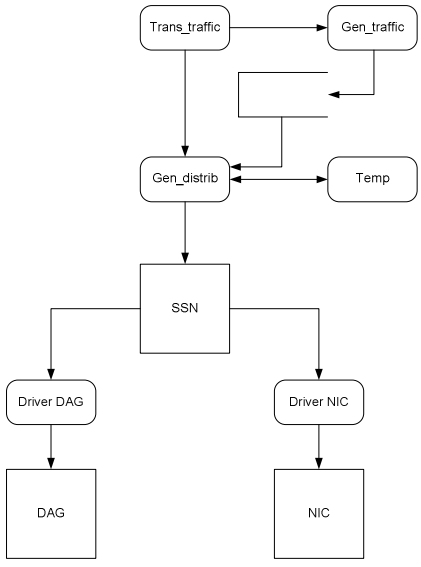


Fig. 1. Arquitectura de referencia del inyector.

VII. Arquitectura de referencia para la monitorización

Los dos objetivos fundamentales del sistema de monitorización son validar el sistema de inyección y obtener una arquitectura genérica.

Para obtener una arquitectura genérica se ha optado por programa un sniffer haciendo uso de la librería libpcap. Esta aplicación puede capturar tanto de la tarjeta de red genérica como de la DAG. Existe un parche que permite, a cualquier aplicación programada usando libpcap, capturar tráfico de las tarjetas DAG. Por tanto, se proponen dos interfaces de monitorización, nuevamente, la tarjeta de red y la tarjeta DAG. En este caso, el interfaz común que tienen es la propia librería.

Por otra parte, para poder validar el sistema de inyección es preciso seguir la distribución del tráfico en el segmento de red. Para ello es importante conseguir realizar un buen marcado temporal al nivel más bajo posible. Si la captura se realiza con la tarjeta de red el marcado se hace a nivel de kernel por parte de la librería. En cambio, las tarjetas DAG realizan un marcado temporal a nivel hardware. Este marcado permite un seguimiento mucho más fiel de la distribución del tráfico de la red. Debido a este marcado, la DAG se plantea como un interfaz de monitorización interesante. Sin embargo se mantiene como interfaz posible la tarjeta de red para mantener la generalidad del sistema.

VIII. Casos de aplicación

Uno de los casos de aplicación fundamentales dentro de los proyectos del grupo de investigación Networking, Quality and Security (NQaS) es la prueba de software de monitorización de tráfico. Actualmente, estas pruebas se realizan utilizando la tarjeta DAG con las aplicaciones de Endace. Anteriormente, estas pruebas se realizaban utilizando el inyector software pktgen como se describe en [8].

En la figura 2 se plantea un posible escenario para experimentación con este sistema. Dicho escenario está compuesto por un equipo de configuración, el sistema de inyección, el sistema de monitorización y un switch que proporciona conectividad. Además, se distinguen dos entornos, el entorno de configuración y el entorno de inyección.

Hasta ahora se ha hecho siempre referencia a los interfaces de inyección y monitorización. Sin embargo, los equipos que contienen los sistemas de monitorización e inyección tienen otro interfaz de red. A través de esos interfaces de red se realiza la configuración de los sistemas desde el equipo de configuración.

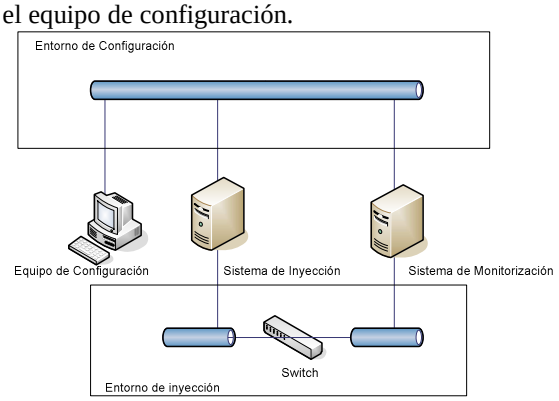


Fig. 2. Escenario para experimentación.

IX. Conclusiones

En este artículo se han propuesto dos sistemas que permiten inyectar y monitorizar tráfico en segmentos de alta capacidad.

El sistema de inyección tiene que conseguir modelar el tráfico que genera inyectando a altas tasas. Esto hace que tenga que tener un buen control del tiempo. En espacio de usuario esto se antoja prácticamente imposible debido a las políticas de planificación con desalojo y consumos computacionales indeseados.

En cuanto al sistema de monitorización tiene que validar el sistema de inyección y, por tanto, seguir la distribución del tráfico en la red. Para ello se necesita un marcado temporal a muy bajo nivel. Dicho marcado se realiza a nivel de kernel e, incluso, a nivel hardware si el interfaz así lo permite, como es el caso de las tarjetas DAG. En un futuro, puede resultar interesante estudiar la modificación de la distribución de tráfico que realiza el propio sistema de captura de tráfico. Para ello además del marcado temporal que realiza este sistema hay que realizar otro marcado a un nivel superior.

Por último, otro objetivo común a ambos sistemas es mantener la generalidad de los mismos. Para ello ambos sistemas se pueden ejecutar en máquinas de propósito general con dos tarjetas de red genéricas. Sin embargo, se mantiene el segundo interfaz de inyección/monitorización debido a las características interesantes que aporta.

Referencias

[1] S. Avallone, S.Guadagno, D. Emma, A. Pescapè, G. Ventre, “A Distributed Multiplatform for Traffic Generation”, Proceedings of International Symposium on Performance Evaluation on Computer and Telecommunication Systems (SPECTS), July 2004, San José, California (USA).

[2] A. Botta, A. Dainotti, A. Pescapè, “Multi-protocol and multi-platform traffic generation and measurement”, INFOCOM 2007 Demo Session, Anchorage (Alaska, USA).

[3] S. Zander, D. Kennedy, G.Armitage, “KUTE – A High Performance Kernel-based Traffic Engine”, Centre of Advanced Internet Architectures (CAIA), Technical Report 050118A. 2005. Melbourne, Australia.

[4] S. Avallone, S.Guadagno, D. Emma, A. Pescapè, G. Ventre. “High performance Internet traffic generators”. The Journal of Supercomputing, 2006, vol. 35, no. 1, p. 5-26. ISSN: 0920-8542

[5] R. Olsson. *"pktgen: the linux packet generator"*. en *11th International Linux System Technology Conference*. 2004.

[6] J. Cleary, et al. *"Design principles for accurate passive measurement"*. *Passive and Active Measurement Workshop PAM*. 2000. Hamilton, New Zealand.

[7] *"TCPDump: the Protocol Packet Capture and Dumper Program"*. 2006. http://www.tcpdump.org/

[8] Muñoz, A., A. Ferro, et al. Ksensor: sistema multiprocesador de análisis pasivo de tráfico a nivel de kernel. VI Jornadas de Ingeniería Telemática, JITEL'07. 2007. Malaga (España).

[9] Mark Crovella and Azer Bestavros. Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes. In Proceedings of SIGMETRICS’96: The ACM International Conference on Measurement and Modeling of Computer Systems., Philadelphia, Pennsylvania, May 1996.

[10] C. Benvenuti. *Understanding Linux Network Internals*, United States of America: O’Reilly, 2005.

Análisis de Seguridad de las Redes Mesh de Sensores en Sistemas Críticos de Control

Cristina Alcaraz, Javier López
Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga, 29071, Málaga, España
{alcaraz,jlm}@lcc.uma.es

***Resumen*—Los sistemas críticos de control representan un componente fundamental para el correcto funcionamiento de muchas de las infraestructuras críticas existentes en nuestra sociedad. Actualmente, estos sistemas incluyen en su diseño infraestructuras y tecnologías de última generación para mejorar los procesos de control, como por ejemplo las redes de sensores. Por ello, varios organismos internacionales están trabajando activamente con el objeto de estandarizar las comunicaciones a este nivel, así como para garantizar la conservación de energía de sus nodos, la coexistencia con las demás redes, y la fiabilidad y seguridad de tales comunicaciones. Desafortunadamente, y tal y como se pone de relieve en este artículo, la seguridad no está totalmente garantizada dado que existen diversas vulnerabilidades asociadas, que ponen en riesgo la funcionalidad general del sistema. En este artículo se realiza un análisis pormenorizado a este respecto, estableciendo además un conjunto de recomendaciones y consideraciones con el fin de mejorar tales especificaciones en los mencionados esfuerzos de estandarización.**

***Palabras Clave*—Sistemas Críticos de Control Críticos, Sistemas SCADA, Redes Mesh Inalámbrica de Sensores.**

I. Introducción

Los sistemas SCADA actuales hacen uso de múltiples y diversas tecnologías para favorecer los procesos de monitorización. De hecho, una de las tecnologías por la que la industria está apostando es la comunicación inalámbrica ya que garantiza los mismos servicios de control que una cableada a un bajo coste de instalación y mantenimiento. Siguiendo en esta misma línea, las *Redes Inalámbricas de Sensores* son también objeto de interés, al proveer atractivos servicios para el control. Es por ello que diversos organismos internacionales están dedicando un esfuerzo nada desdeñable a la estandarización de sus respectivas comunicaciones.

Debido a la criticidad de los sistemas de control y de los sistemas monitorizados, es necesario analizar todos los posibles ataques asociados a estos estándares de comunicación, y asegurar seguridad de los datos críticos, disponibilidad de recursos y protección de ciertos servicios de operación, tal como: comandos/órdenes o lecturas/alarmas. Además, en este tipo de análisis entra en juego el contexto de aplicación, las acciones intencionadas de los propios miembros de la organización y de los miembros ajenos a ésta. Como resultado, un conjunto de recomnedaciones se han propuesto para mejorar sus especificaciones y mitigar cualquier tipo de efecto en cascada sobre el sistema monitorizado.

El artículo está organizado en cinco secciones: la sección II introduce el concepto de sistema crítico de control y presenta la tecnología de las redes de sensores para el control de las infraestructuras críticas. En la sección III se introducen los estándares de comunicación, profundizando en la seguridad de los mismos, que pasa a ser analizada y discutida en la sección IV. Finalmente, en la sección V se encuentran las conclusiones y trabajo futuro.

II. Sistemas SCADA y tecnologías

Un sistema SCADA [1] es un sistema de control cuya labor es la de monitorizar otras infraestructuras críticas pertenecientes principalmente al sector industrial. Las características implícitas y críticas de estos sistemas hacen que un sistema SCADA requiera funcionalidad del sistema, disponibilidad de sus recursos y, sobre todo, seguridad dado que son muy vulnerables a multitud de ataques, la mayoría derivados de acciones maliciosas o negligentes.

El núcleo principal de estos sistemas lo forma el "Centro de Control", en el que los operarios pueden conocer en tiempo real el estado de las infraestructuras monitorizadas simplemente observando datos o alarmas recibidas por las subestaciones remotas instaladas de manera jerárquica en todo el área. La gestión de estas subestaciones se puede realizar desde cualquier localización geográfica y a través de diversas infraestructuras de comunicación, como puede ser la inalámbrica. De hecho, varias organizaciones ([2], [3], [4]) están trabajando activamente en la estandarización de éstas, las cuales contemplan a su vez las redes de sensores [5] como un perfecto elemento de control. Este hecho se debe principalmente por las características inherentes de sus nodos sensores, cuyas mediciones se corresponden con las condiciones físicas y reales de las infraestructuras monitorizadas. Obviamente, una red de sensores en un sistema SCADA debe enviar tales datos al centro de control para que los operarios puedan ser capaces de interpretarlos. De forma similar, estas redes pueden ofrecer a los operarios ciertos servicios de consulta en tiempo real, detectar situaciones anómalas y generar alarmas con el fin de resolver el problema dentro de un periodo de tiempo límite. Asimismo, los nodos sensores pueden ofrecer autonomía, independencia y auto-configuración para funcionar en cualquier área de desarrollo y en cualquier contexto de aplicación.

Sin embargo, distribuir una red de sensores como parte esencial de un sistema de control supone tener en cuenta algunos aspectos relacionados con la conservación de energía, fiabilidad de las comunicaciones al existir una alta probabilidad de interferencias industrial, coexistencia con otras redes de comunicación, restricciones hardware y softwarre, y por supuesto seguridad. Con respecto a este último, es importante comentar que existen algunos avances en la utilización de primitivas criptográficas (de clave simétrica y pública), funciones hash, y mecanismos de negociación y distribución de

claves [6].

III. Estándares de Comunicación y su Seguridad

Los estándares ZigBee PRO [4], WirelessHart [7] e ISA100.11a [8] están basados de IEEE 802.15.4-2006 [9], cuyos nodos tramisten a baja frecuencia y tienen limitadas capacidades hardware y software, así como también de energía. Sus dispositivos pueden funcionar a 2.4GHz a 250 kbps o 868-915MHz a 20 kbps, con 15 canales de transmisión, y cuya capa de enlace controla los accesos al medio mediante CSMA-CA. Además, provee soporte para AES-128bits y gestiona una lista de control de acceso (ACL, *Access Control List*), donde se mantiene el ID del nodo a comunicar, la política de seguridad a utilizar, una clave de 128 bits, un vector inicial y un contador. En el caso de que el nodo no esté en dicha lista, su mensaje debe ser rechazado o tiene que pasar otros mecanismos de autentificación.

A. ZigBee PRO

ZigBee PRO, especificada en ZigBee-2007, tiene como objetivo proporcionar coexistencia y control en redes de comunicación mesh y redes de muchos-a-uno. Su arquitectura de red está basada en tres nodos principales: un gateway (entidad de mayor confianza), routers y los nodos sensor finales. Dicha red provee varios servicios, como son: un "gestor de enlace asimétrico" para configurar aquella ruta con mejor calidad simétrica entre pares de nodos, "agilidad de la frecuencia" para analizar las interferencias u obstáculos en canal y cambiar de canal si hace falta, "rutas compartidas de muchos-a-uno" donde se mantiene una tabla de enrutamiento con una única entrada hacia el gateway, y "enrutamiento origen" para recordar la ruta de vuelta desde el gateway al nodo origen. También, se controla los conflictos de identidad mediante el uso de "direccionamiento estocástico", donde apriori se asigna a cada nodo nuevo una dirección aleatoria y en caso de conflicto activar un mecanismo de resolución de identidades. Por último, ZigBee PRO provee dos modos de seguridad: "seguridad estándar" y "seguridad alta", y ambas son mantenidas por el gateway.

En el modo de "seguridad estándar" se manejan dos claves: clave de enlace (Cenl) y de red (Cred). La Cenl (única y opcional) es compartida entre pares de nodos, y es usada para cifrar los mensajes en la capa de aplicación. En cambio, la Cred es una clave usada para cifrar las comunicaciones a nivel de red y es compartida por todos los dispositivos del sistema. Esta clave puede ser actualizada por el gateway a través de un mensaje en difusión cifrada con la antigua Cred. Cuando un nodo se une a la red puede adquirir la Cred de dos maneras, bien haciendo uso de la Cenl preconfigurada en el nodo para cifrar la Cred, o bien, recibirla desde el gateway en claro. Obviamente, este último caso no es muy aconsejable en entornos críticos.

En cambio, en el modo de "seguridad alta" se incluye una clave más al conjunto anterior: la clave maestra (Cmaestra). Esta clave es preconfigurada en el nodo para generar la Cenl aplicando el algoritmo Symmetric-Key Key Exchange (SKKE). Una vez generada la Cenl, el gateway transmite la Cred cifrada con la Cenl al nodo correspondiente. Esta clave de red es actualizada de manera periódica, incluso cuando los nodos son excluidos de la red. El principal problema asociado a este modo es la alta sobrecarga de memoria, sin embargo, esto garantiza seguridad en aplicaciones críticas.

B. WirelessHart

WirelessHart, especificado como parte de [10], define un protocolo de red mesh para el control de automatización industrial manteniendo máxima compatibilidad con las tecnologías ya existentes de HART. Su arquitectura de red está compuesta de nodos sensores adheridos a dispositivos de campo, dispositivos portables (PDAs, móviles, etc.), gateways para la interconexión entre sistemas de comunicación y un gestor de red. Dicho gestor, el cual podría estar integrado en el propio gateway, establece la configuración de red y define las tablas de enrutamiento de los nodos, así como también, planifica las comunicaciones entre dispositivos.

Su capa capa física está bajo [9], sin embargo, éste define su propia capa de enlace estableciendo tiempos fijos de sincronización mediante TDMA/CSMA. Controla las altas interferencias en los canales de comunicación aplicando los métodos "blacklisting" (incluir en una lista aquellos canales con alto índice de ruido) y "hopping" (cambiar de canal de radio frecuencia). Con respecto al enrutamiento de mensajes, el propio gestor de red establece las diversas rutas redundantes asociadas a un nodo de la red. La actualización de dichas tablas se realiza por cada nueva integración y se procede en todos los nodos de red involucrados en la comunicación. También, WirelessHart ofrece seguridad tanto a nivel de enlace como a nivel de red. En ambos, se hace uso de cuatro tipos de claves diferentes: clave pública (Cpub), usada para generar en la fase de despliegue el código de integridad de mensajes (MIC) de la capa de enlace. Clave de red (Cred), compartida por todos los dispositivos y utilizada para generar el MIC en la capa de enlace. Clave de unión (Cunión), única para cada nueva integración y es usada para generar el MIC en la capa de red y para cifrar el mensaje de nueva unión. Clave de sesión (Csesión), única entre un nodo de la red y el gestor de red, y es utilizada para cifrar los mensajes. En lo que respecta a la generación del MIC, ésta está basada en el modo CCM* junto con AES-128bits y tomando como parámetros: la cabecera del paquete sin cifrar y su cuerpo, una clave de 16 bytes (Cpub si el nodo es nuevo o Cred si ya existe) y un nounce único de 13 bytes.

Antes de que un nodo nuevo se integre en la red, éste debe ser preconfigurado con la ID de la subred a la que tiene que unirse, la Cpub y la Cunión. Cuando el nodo es posicionado en la red, éste debe hacerse conocer públicamente mediante un mensaje de nueva unión, junto con el MIC de la capa de enlace cifrado con la Cpub y el MIC de la capa de red cifrado con la Cunión. Una vez que este mensaje es recibido por el gestor de red, éste lo autentifica con su clave privada y genera una Csesión única. Tras la generación, el gestor trasmite la Csesión y la Cred a la nueva incorporación cifrado con la Cunión. En paralelo, el gestor prepara la nueva planificación y la tabla de enrutamiento, y ambas, son transmitidas a todos los nodos de la red. En el caso de que un nodo sensor ya existente en la red desee enviar datos al gestor de red, éste deberá autentificarse con la Cred y el paquete es cifrado con la Csesión. Por último, es importante comentar que todas estas claves no son actualizadas durante toda la vida útil de un nodo sensor (entre 5 y 10 años) [11].

C. ISA100.11.a

ISA100.11.a es un estándar recientemente validado y está pensado para ser un estándar abierto. Su especificación está planteada para ser funcional en redes mesh o en estrella cuyos nodos son baja complejidad. Está enfocado para ser aplicados en aplicaciones industriales, garantizando conservación de energía, escalabilidad, interoperabilidad, fiabilidad en las comunicaciones y seguridad. Como puntualización, al no encontrarse este estándar aún disponible, no se ha sido posible realizar el estudio de seguridad.

IV. Análisis de Seguridad de ZigBee PRO y WirelessHart

Considerando como taxonomía base de amenazas la propuesta por Tsao et al. en [12], varios ataques han sido identificados en los respectivos estándares.

A. Amenazas Zigbee PRO

Debido al contexto crítico de los sistemas de control, el análisis se enfocará sobre el modo de "seguridad alta". De hecho, varios ataques contra la confidencialidad han sido ya identificados, como por ejemplo: *exposición deliberada* y *control de acceso remoto*, donde un operador del sistema conoce y preinstala la Cmaestra en un nodo sensor para pasar los mecanismos de autentificación en el gateway y unirse a la red como si éste fuese un nuevo nodo legítimo. Similarmente, existe ataque de *escucha del canal* cuando un operador conoce la Cmaestra y es capaz de interpretar las transacciones de SKKE entre dos nodos de la red con el fin de generar la Cenl de ambos. En dicho momento, el adversio puede tener la capacidad de deducir la Cred, y por lo tanto, leer cualquier mensaje en el canal. Además, un malicioso miembro ajeno de la organización podría ser también capaz de deducir la Cenl observando el contenido de los mensajes, por lo que se recomienda actualizarla de manera frecuente. Además, para mitigar los tres ataques anteriores sería conveniente establecer rigurosas políticas de seguridad, llevar a cabo regulares auditorías (p. ej. NIST SP800-53 revisión 2) y hacer uso de mecanismos de autentificación.

Un adversario es capaz también de deducir la topología de la red simplemente observando el tráfico de la red, el cual va dirigido siempre hacia el gateway. Este hecho podría capacitar al adversario a aproximarse a la localización física del gateway para realizar otros ataques futuros. Una solución sería mantener una tabla con varias entradas, sin embargo, esto no es factible, ya que las tablas tienen una sóla entrada. Por consiguiente, sería necesario actualizar dichas tablas de manera periódica. Con respecto a los *ataques físicos*, éstos dependerán del tipo de protección del área de trabajo y del tipo de acceso a la misma. Lo ideal sería configurar mecanismos de protección en todo el perímetro de distribución haciendo uso de procedimientos de autentificación para el acceso, así como el uso cámaras de video vigilancia. Por otro lado, es necesario seguir avanzando en el diseño de la plataforma hardware de los nodos para que sean resistentes frente ataques físicos.

No existe ataque de *manipulación de datos* al transmitirse el MIC junto con el mensaje cifrado. Tampoco existe ataque de *reenvío de mensajes* al utilizarse un contador único en cada envío. Sin embargo, y aunque no exista ningún nodo específico que reconfigure las tablas de enrutamiento, existe la posibilidad de que varios nodos maliciosos en la red puedan mentir sobre la calidad de sus rutas en el momento de determinar cúal es la ruta con mejor calidad simétrica al gestor de enlace asimétrico. Luego, existe ataque de *falsificación de rutas*. También, puede existir ataque *sybil* cuando la clave Cenl entre dos nodos es deducida por haber comprometido la Cmaestra, por lo que se recomienda realizar periódicas actualizaciones de ésta. Por el contrario, no existen ataques *sinkhole* y *wormhole* al gestionarse una ACL con todos los vecinos de confianza. En cambio, los ataques de *inundación* de paquetes y de *reenvío selectivo* son controlados al gestionarse tablas de enrutamiento de una sóla entrada, y por lo tanto, un sólo nodo vecino deberá recibir el mensaje. Tampoco existe sobrecarga del canal y *obstrucción del medio* al utilizarse la técnica de agilidad de frecuencia.

Existe ataque *blackhole* cuando un nodo supuestamente legítimo de la red no reenvía los mensajes. Es posible mitigar este ataque enviando el mensaje por varias rutas a la vez o seleccionando dinámicamente el siguiente nodo de entre un conjunto de candidatos. Sin embargo, estas soluciones entran en conflicto con la propia definición del protocolo, por lo que se deberá tener en cuenta para mejorar su especificación. También, es posible *aislar la red* cuando el gestor de enlace asimétrico recibe datos falsos de los nodos vecinos que mienten sobre la calidad de su enlace para anular todas las posibles rutas a tomar. Esto supone que la mayoría de nodos de la red, y al menos uno en cada ruta, están comprometidos. Lo cual puede ser resuelto mediante frecuentes procesos de mantenimiento y de inspección de la red.

B. Amenazas en WirelessHart

En WirelessHart pueden existir varios tipos de amenazas, como: *exposición deliberada* y *control de acceso remoto* donde un operador quiere ganar ciertos accesos al sistema preinstalando la Cpub y Cunión en un nodo supuestamente legítimo de la red. Por ello, es necesario implantar fuertes políticas de seguridad, que incluyan rigurosos mecanismos y procedimientos de autentificación y auditoría. Por otro lado, como no se cifra al completo los mensajes intercambiados en el canal de comunicaciones y la actualización de las tablas de enrutamiento de los nodos dependen de la frecuencia de unión de nuevos nodos a la red, puede existir un ataque de *análisis del tráfico*. Una solución sería actualizar periódicamente las tablas de enrutamiento sin que ello suponga esperar nuevas y futuras (quizás lejanas) incorporaciones. También, existe posibilidad de que se lleve a cabo un *ataque físico* si el área de desarrollo es fácilmente accesible o está totalmente desprotegida, por lo que se recomienda securizar el medio y proteger la estructura hardware de los nodos. Igualmente, un adversario puede ser capaz de deducir las credenciales de seguridad de un nodo sensor al no requerirse frecuentes actualizaciones de éstas durante toda su vida útil. Por lo que si un atacante deduce las Csesión entre un nodo y el gestor de red, éste es capaz de leer el contenido de todos los mensajes cifrados con dicha clave. Obviamente, la solución sería actualizar las credeciales de seguridad de manera periódica.

Aunque no existe ataque de *manipulación de datos críticos* por enviar los mensajes junto con el MIC generado con la Cred, existe ataque de *falsificación de rutas* cuando el gestor de red es comprometido por algún miembro de la organización

con determinados accesos al sistema. Una forma de evitar este ataque es securizando todos los puntos de accesos al sistema SCADA y aprovechar las capacidades hardware y software del gestor para configurar mecanismos de autentificación potentes, además de realizar frecuentes procedimientos de auditorías. Por el contrario, no existe ataque *sybil* durante el proceso de únion al autentificarse con la Cpub. Sin embargo, si es posible realizar ataques sybil cuando la Csesión entre un nodo y el gestor es comprometida. Para mitigar este ataque lo ideal sería realizar frecuentes actualizaciones de la Csesión y que el propio gestor de red sea capaz de controlar la recepción de los mensajes de una supuesta y misma identidad.

No existe ataque de *reenvíos de mensajes*, ya que se usa un nounce de 13 bytes para la generación del MIC. Asimismo, no existe ataque de *obstrucción del medio* al utilizarse la técnica hopping y el uso de rutas redundantes. Tampoco existe ataque de *inundación* de paquetes al establecerse rutas concretas hacia el gestor de red. Por el contrario, existe ataque *reenvío selectivo* al mantenerse una tabla de enrutamiento con rutas redundantes, por lo que sería conveniente enviar dicho mensaje por varias rutas diferentes. También, existe ataque *blackhole* cuando un nodo supuestamente legítimo de la red recibe datos y no los reenvía al nodo siguiente de la tabla de enrutamiento. Para evitar dicho ataque es necesario modificar directamente la definición del protocolo para permitir el envío de paquetes por múltiples rutas o mediante una selección dinámica del siguiente nodo.

Puede existir un ataque *sinkhole* y *wormhole* cuando un nodo (o varios) de la red transmite información falsa al gestor de la red para modelar las tablas de enrutamiento a su beneficio. Esto se puede evitar estableciendo una política de aislamiento de nodos maliciosos usando un umbral específico sobre la cantidad de tráfico o recibir la información sólo del nodo vecino fiable mediante el uso de algún mecanismo de confianza. Igualmente, pueden surgir *aislamientos de la red* cuando ciertos ataques de denegación de servicios aparecen en la interfaz entre el centro de control y la red de sensores. En este caso un operador malicioso podría sobrecargar al gateway con múltiples órdenes/comandos mediante el uso de algún protocolo SCADA (p.ej. Modbus/TCP [13]). Una forma de evitarlo sería implementar mecanismos automatizados e inteligentes que gestionen la frecuencia de envío por operador favorenciendo los procesos de auditorías futuros. Otra forma de aislar la red se consigue generando altas interferencias en el canal con el fin de bloquear los 15 canales de frecuencias. Este hecho incapacita a los nodos a transmitir por ningún canal. En este caso, se aconseja monitorizar el medio de distribución para conocer el origen del fenómeno que genera las interferencias y quitar de la lista blacklisting todos aquellos canales ya disponibles.

V. Conclusiones y Trabajo Futuro

El proposito de este artículo ha sido realizar un profundo análisis de seguridad de los estándares de comunicación inalámbricos aprobados recientemente, donde es posible observar en la tabla I que la mayoría de los ataques son generalmente originados por los propios (ex-)miembros del sistema. Bajo estas condiciones, se recomienda actualizar las tablas de enrutamiento y credenciales de seguridad, no sólo frecuentemente, sino cuando un operador abandona (temporalmente o definitivamente) la organización. Igualmente, se hace necesario diseñar e implementar mecanismos automatizados e inteligentes capaces de explorar todas las actividades desarrolladas, así como también, establecer rigurosas políticas de seguridad, y realizar frecuentes auditorías y mantenimiento. Todas estas recomendaciones y muchas otras han sido comentadas a lo largo de este artículo. Por último, es importante comentar que el análisis de seguridad de ISA100.11.a se ha propuesto como trabajo futuro al no estar aún disponible.

Tipos de Ataques	ZigBee PRO		WirelessHart	
Exposición Deliberada	✓	I	✓	I
Escucha del canal	✓	A	✓	A
Análisis del Tráfico	✓	A	✓	A
Acceso y Control Remoto	✓	I	✓	I
Ataque Físico	✓	A	✓	A
Manipulación de Datos Críticos	x	-	x	-
Falsificación de Rutas	✓	I	✓	I
Reenvío de Mensajes	x	-	x	-
Inundación	x	-	x	-
Reenvío Selectivo	x	-	✓	I
Blackhole	✓	I	✓	I
Sybil	✓	A	✓	A
Sinkhole	x	-	✓	I
Wormhole	x	-	✓	I
Obstrucción del Medio	✓	A	✓	A
Aislamiento de la Red	✓	I	✓	I

Tabla I
ANÁLISIS DE SEGURIDAD DE ZIGBEE PRO Y WIRELESSHART. MIEMBRO DEL SISTEMA (I), MIEMBRO AJENO (E) Y AMBOS (A)

Agradecimientos

Este trabajo ha sido financiado por CRISIS (TIN2006-09242) y ARES (CSD2007-00004), agradeciendo a Rodrigo Román sus constructivos comentarios.

Referencias

[1] C. Alcaraz, G. Fernández, R. Román, A. Balastegui, J. López, "Secure Management of SCADA Networks", New Trends in Network Management, CEPIS, vol. IX, no. 6, pp 22-28, 2008.
[2] ISA100 "Wireless Systems for Automation", http://www.isa.org/Content/NavigationMenu/Technical_Information/ASCI/ISA100_Wireless_Compliance_Institute/ISA100_Wireless_Compliance_Institute.htm, 2009.
[3] HART Communication, http://www.hartcomm2.org, 2009.
[4] ZigBee Alliance, http://www.zigbee.org/, 2009.
[5] J. Yick, B. Mukherjee, D. Ghosal, "Wireless sensor network survey", Computer Networks Journal, vol 52, num 12, pp 2292–2330, Elsevier, 2008.
[6] R. Roman, C. Alcaraz, N. Sklavos. "On the Hardware Implementation Efficiency of Criptographic Primitives", Cryptology and Information Security Series, vol 1, pp 285–305, 2008.
[7] S. Jianping, H. Song, K. Mok,C. Deji, M. Lucas, M. Nixon, "WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control", RTAS apos'08, vol 22, num 24, pp 377–386, 2008.
[8] ISA100.11a, http://www.isa.org/MSTemplate.cfm?MicrositeID=1134&CommitteeID=6891, 2009.
[9] IEEE 802.15.4-2006, http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf, 2006–2009.
[10] HART Communication Foundation, http://www.hartcomm2.org/hart_protocol/wireless_hart/hart7_overview.html, 1993–2009.
[11] D. Nilsson, T. Roosta, U. Lindqvist, A. Valdes, "Key management and secure software updates in wireless process control environments", WiSec'08, pp 100–108, ACM, 2008.
[12] T. Tsao, R. Alexander, M. Dohler, V. Daza, A. Lozano, "A Security Framework for Routing over Low Power and Lossy Networks", http://tools.ietf.org/id/draft-tsao-roll-security-framework-00.txt, 2009.
[13] Modbus-IDA. "The Architecture for Distributed Automation", http://www.modbus.org/, 2005–2009.

Desarrollo de Aplicaciones Basado en Patrones de Seguridad

Daniel Serrano[1], José F. Ruíz[2], Antonio Maña[2] y Antonio Muñoz[2]

1 CINNTA, España
dserrano@cinnta.es
2 Universidad de Malaga, España
{joseruiz,amg,amunoz}@lcc.uma.es

Abstract- Actualmente los enfoques existentes para el desarrollo de software presentan carencias en la integración de la seguridad de los mismos. Normalmente, esto se debe a la complejidad del software y a que se debe de disponer de la habilidad necesaria para la integración de las soluciones de seguridad modernas. En este artículo presentamos el proyecto SERENITY, el cual consiste en un framework que ofrece una solución al problema anteriormente planteado. SERENITY se basa en establecer una separación entre el desarrollo de las soluciones de seguridad y el desarrollo del software seguro que utiliza estas soluciones. Nuestro enfoque está dirigido para ser usado en los nuevos escenarios de inteligencia ambiental, grids,etc. A lo largo de este artículo presentamos el desarrollo de una aplicación de seguridad basada en el framework de SERENITY, presentando un Interfaz de Programación de Aplicación (del inglés API) especialmente desarrollado para SERENITY.

Palabras clave: seguridad y dependabilidad, Inteligencia Ambiental, patrones de seguridad, desarrollo.

I. INTRODUCCIÓN

La programación ubicua, la inteligencia ambiental (AmI) y la computación autónoma, presentan nuevos retos, especialmente desde el punto de vista de la seguridad y la dependabilidad (S&D).

La realización del concepto de la Inteligencia Ambiental engloba muchas acciones importantes. La mas importante que hay que cumplir es tener un soporte adecuado de seguridad. En este artículo presentamos un modelo conceptual de nuestra solución para la construcción de sistemas de seguridad para ambientes inteligentes (AmI), tomando como base el concepto de los patrones de Seguridad y Dependabilidad (S&D) para obtener una representación precisa de soluciones y mecanismos S&D válidos.

Este artículo muestra un Interfaz de Programación de Aplicaciones (API) para el desarrollo de aplicaciones seguras para ser usado en estos nuevos escenarios. Nuestra solución se basa en los resultados del Proyecto SERENITY [1].

SERENITY ofrece dos frameworks. En primer lugar, un framework, llamado SERENITY Development-time Framework, dirigido a ayudar a los desarrolladores de aplicaciones seguras. Y en segundo lugar, un framework que proporciona soporte a las aplicaciones desarrolladas siguiendo esta propuesta, siempre en tiempo de ejecución, al que hemos llamado SERENITY Run-time Framework. Una de las principales motivaciones de esta propuesta es que resulta muy complicado para los programadores desarrollar software seguro para los nuevos paradigmas tecnológicos. Usando la API que se propone, los desarrolladores pueden crean aplicaciones seguras que son mantenidas por los frameworks de SERENITY para el suministro de mecanismos de seguridad y dependabilidad (S&D). Además, este artículo presenta los resultados de la implementación de una aplicación de mensajería instantánea usando la API implementada en Java que se ha visto anteriormente.

El resto del documento está organizado de la siguiente forma: La siguiente sección ofrece una revisión de los trabajos relacionados. La sección 3 presenta el concepto de SERENITY de las soluciones de S&D , mostrando cómo son capturadas a través del concepto de los patrones S&D. Además, la sección 3 presenta los dos framework SERENITY. La sección 4 presenta las ideas principales de este artículo, presentando nuestra API. La sección 5 presenta el escenario. Y por último, resumimos nuestro trabajo y ofrecemos conclusiones a los lectores.

II. TRABAJOS RELACIONADOS

El trabajo presentado en este artículo está relacionado con el desarrollo de aplicaciones seguras basadas en el uso de patrones de seguridad. A lo largo de esta sección de trabajos relacionados ofrecemos una revisión del uso de patrones y en concreto de patrones de seguridad en el desarrollo software.

El concepto de patrón de seguridad fue presentado por primera vez por Yoder y Barcalow [4] en 1997. Después de esto, se han publicado varios trabajos [5], [6] sobre patrones y su aplicación en la seguridad software. El punto débil de los patrones de seguridad es que no poseen un lenguaje preciso. SERENITY soluciona esta debilidad ofreciendo un lenguaje que permite la descripción semántica de patrones de seguridad. SERENITY pone estos patrones de seguridad al servicio de los desarrolladores gracias a las librerías on-line. Han habido grandes esfuerzos para crear librerías de patrones de seguridad. Estas librerías definen una jerarquía y distribución de patrones. Algunos ejemplos de colecciones de patrones se pueden encontrar en [7], [8], [9].

Con respecto al uso de patrones de seguridad en la inteligencia ambiental (AmI), podemos encontrar propuestas basadas en aplicaciones sostenidas por patrones que ofrecen control y seguridad en un gran número de escenarios, tal como muestran los trabajos [10-12]. La evolución de estas aproximaciones se ha visto incrementada desde la primera vez que fueron usadas [13-15]. La diferencia entre SERENITY y estas propuestas consiste en que las aplicaciones desarrolladas usando patrones de seguridad tradicionales son sostenidos por los mismos patrones sea cual sea el contexto en el que se encuentren. De esta manera es imposible cambiar el patrón de seguridad que se está usando en una aplicación en tiempo de ejecución, y por tanto no se puede reaccionar ante cambios de contexto, como ocurre en los entornos AmI.

III. INTRODUCCIÓN A SERENITY

Como mencionamos anteriormente, la contribución que se presenta en este artículo es un Interfaz de Programación de Aplicaciones (API) para desarrollar aplicaciones SERENITY. Este tipo de aplicaciones son aplicaciones [16] que trabajan en entornos AmI desarrolladas y sostenidas en tiempo de ejecución por el marco de trabajo de SERENITY. Para facilitar la comprensión de la contribución, esta sección presenta el Proyecto SERENITY [1].

El Proyecto SERENITY ofrece un marco de trabajo para el tratamiento automático de la Seguridad y Dependabilidad (S&D) en escenarios AmI. Para hacer esto, el Proyecto SERENITY tiene dos puntos clave: (i) capturar la habilidad de los ingenieros de seguridad para permitir el procesamiento automático y (ii) ofrecer los medios para realizar una monitorización en tiempo de ejecución tanto de los mecanismos de seguridad como de dependabilidad. Estas dos claves se han logrado medio de un conjunto de componentes descritos a continuación.

A. *Artefactos S&D*

Los artefactos S&D son un conjunto de artefactos software usados para representar e implementar las soluciones S&D. SERENITY ofrece un conjunto de artefactos para proporcionar el desarrollo de soluciones complejas a diferentes niveles de abstracción. Existen tres niveles de abstracción, partiendo de las descripciones abstractas de soluciones S&D hasta las implementaciones totalmente desarrolladas de esa solución. Tenemos un tipo de artefacto S&D específico para describir una de estas soluciones en cada nivel de abstracción:

- En el nivel mas abstracto, nuestras soluciones se describen en términos de conjuntos de Propiedades S&D e interfaces. A este nivel nos interesa más que encontrar una solución particular S&D, encontrar soluciones S&D utiles para solucionar diversos problemas de forma general. El artefacto de nuestra jerarquía que captura esta información es la Clase S&D (Fig. 1).
- En el nivel intermedio de abstracción nos encontramos con ciertas soluciones S&D que se detallan con sus operaciones. En este nivel, las soluciones S&D se describen usando un artefacto S&D que se llama Patrón S&D (ver Fig. 1). Es importante decir que cualquier Patrón S&D está asociado a una Clase S&D.
- El nivel mas bajo de las soluciones S&D se describen por medio de dos artefactos S&D. Uno de ellos es el código de la solución S&D implementada , al cual llamamos Componente Ejecutable (ver Fig. 1). El otro corresponde con la descripción del código actual, esto es: la descripción del Componente Ejecutable. Este último artefacto se llama Implementación S&D. Cada Implementación S&D representa la implementación de un Patrón S&D (ver Fig. 1).

La Fig. 1 muestra todos estos artefactos S&D y sus relaciones. Esta figura muestra también como una solución S&D se describe usando los artefactos SERENITY. En el nivel más alto de la abstracción la solución S&D se representa usando un concepto: Cifrar. En el nivel de Patrón S&D hay dos tipos diferentes de soluciones S&D que pertenecen a la Clase S&D de cifrado: DES y AES. Por último, en el nivel mas bajo de abstracción hay tres implementaciones de estos Patrones S&D. Existen dos implementaciones del Patrón S&D DES y una implementación del Patrón S&D AES. Para cada implementación existe un artefacto Implementación S&D y un Componente Ejecutable.

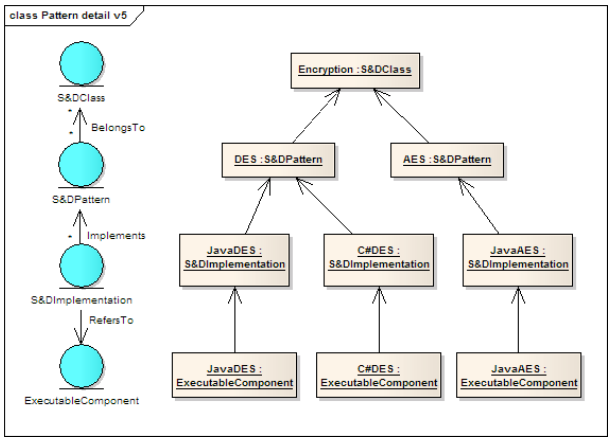


Figura 1.

B. *Framework de desarrollo*

El framework de desarrollo de SERENITY (SDF) ofrece:

- El desarrollo de soluciones S&D por medio de los artefactos presentados anteriormente.
- El desarrollo de aplicaciones seguras mediante el uso de SERENITY. Estas aplicaciones seguras son procesadas o soportadas en SERENITY para satisfacer sus requisitos.

El SDF usa repositorios on-line que tienen artefactos S&D. Por una parte, los expertos de seguridad usan estos repositorios online para almacenar las soluciones S&D que desarrollan. Una descripción más detallada del desarrollo de aplicaciones basado en estos repositorios online se puede encontrar en [17].

C. *Framework de tiempo de ejecución*

SERENITY también ofrece un marco de trabajo para el tiempo de ejecución llamado SERENITY Run-time Framework (SRF). El SRF proporciona soporte a las aplicaciones de SERENITY en tiempo de ejecución, coordinando soluciones S&D y monitorizando el contexto del sistema.

Las aplicaciones SERENITY incluyen referencias a estas soluciones S&D de SERENITY. El SRF procesa estas referencias en tiempo de ejecución e instancia Componentes Ejecutables de acuerdo a las solicitudes realizadas por las aplicaciones. Una vez que el SRF instancia un Componente Ejecutable, este, le manda un manejador a la aplicación. Con este manejador las aplicaciones pueden acceder a los Componentes Ejecutables. Mientras los Componentes Ejecutables están funcionando y siendo accedidos por aplicaciones SERENITY, el SRF monitoriza y controla su correcto funcionamiento.

IV. API JAVA PARA EL DESARROLLO DE APLICACIONES SERENITY

El desarrollo de aplicaciones SERENITY se basa en la inclusión de referencias a artefactos S&D. Esta sección describe una API Java para el manejo de referencias a soluciones S&D. Para cada referencia a un artefacto S&D, la aplicación incluye una petición al SRF. En estas peticiones las aplicaciones expresan que artefacto S&D necesitan.

Los desarrolladores de aplicaciones SERENITY deben trabajar con dos interfaces. Primero, el interfaz SRF que se

usa para solicitar los artefactos. Y el segundo, el interfaz del artefacto S&D solicitado que implementa el Componente Ejecutable. Este segundo interfaz se usa para acceder a las funcionalidades de las soluciones S&D implementados en los Componentes Ejecutables. Actualmente, estos interfaces se han implementado por medio de sockets.

Para simplificar el desarrollo de aplicaciones SERENITY, esta sección presenta una API para Java [18]. Esta API ayuda en la tarea de acceder tanto al SRF como a los Componentes Ejecutables.

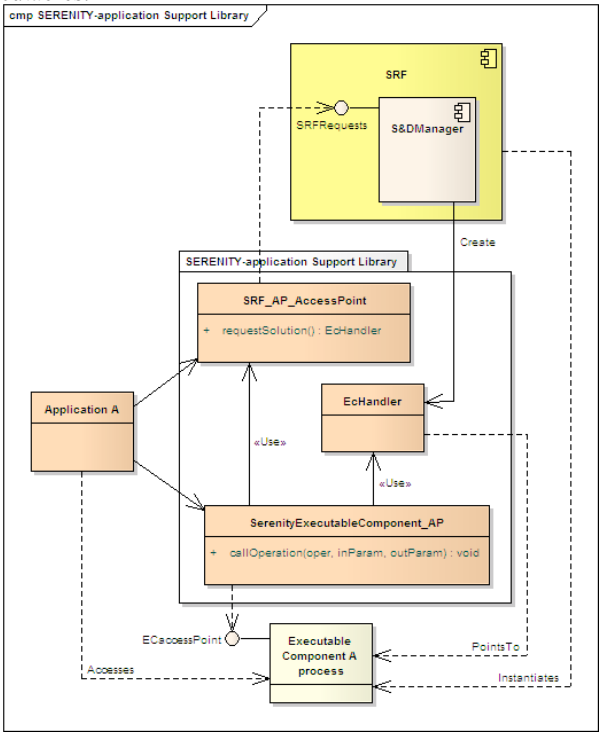


Figura 2.

Además de eso, el desarrollo de aplicaciones SERENITY consiste en seguir una serie de pocos pasos que van desde la conexión con el SRF al uso de un Componente Ejecutable (o un conjunto de Componentes Ejecutables).

El primer paso es crear la conexión entre la aplicación SERENITY y el SRF donde se ejecutará esta. Desde el punto de vista de la aplicación el SRF está encapsulado en la clase *SRF_AP_AccessPoint*

Una vez que se ha creado la conexión con el SRF, la aplicación puede solicitar artefactos S&D. Los Componentes Ejecutables están encapsulados en objetos de la clase *SerenityExecutableComponent_AP*. La aplicación debe declarar un nuevo objeto de esta clase y usar el método constructor para expresar el artefacto S&D que va a usar. Notar que, uno de los parámetros del método constructor es una referencia al SRF, esto es, un objeto de la clase *SRF_AP_AccessPoint*. Después de la creación de los Componentes Ejecutables la aplicación puede acceder a sus funcionalidades. Para hacer esto, la clase *SerenityExecutableComponent_AP* ofrece un método llamado *callOperation*. Este método se usa para acceder al interfaz de los Componentes Ejecutables.

La Fig. 2 muestra un diagrama de clases con las clases y métodos ofrecidos por la API de desarrollo de aplicaciones SERENITY. La API está incluida en un paquete llamadlo Librería de Apoyo a las aplicaciones SERENITY. Aparte de las clases anteriormente mencionadas existen otras clases que la API usa internamente. La clase *ECHandler* encapsula la información de los Componente Ejecutables. Los objetos de esta clase son creados por el SRF como resultado de una petición de la aplicación. Estos son controlados por objetos de tipo *SerenityExecutableComponent_AP* para poder acceder a ellos.

V. Escenario: Servicio de Mensajería Instantanea Dependiente del Contexto

Esta sección muestra un escenario que nos servirá como ejemplo para ver la utilidad de la API mostrada en este artículo. En este escenario existen dos aplicaciones de mensajería instantánea. Estas usan soluciones S&D para cifrar y descifrar los mensajes que se intercambian. Las aplicaciones están ejecutándose en diferentes nodos y están utilizando SRFs. Para simplificar lo máximo el prototipo, las soluciones S&D usadas incluyen mecanismos de criptografía pero no tienen métodos para la comunicación entre ellas. La Fig. 3 presenta un diagrama de despliegue que muestra la arquitectura del prototipo desarrollado. Posee dos nodos. Cada nodo contiene una aplicación SERENITY, en este caso la aplicación de mensajería instantánea, un SRF y un Componente Ejecutable.

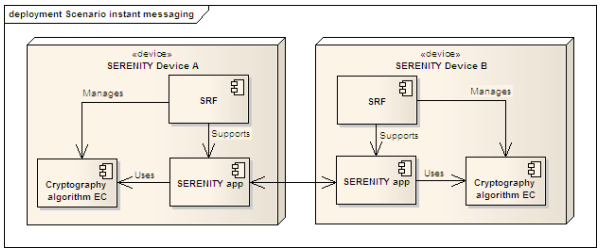


Figura 3.

A. Elementos del escenario

Para construir este escenario hemos desarrollado tres soluciones S&D de criptografía: el algoritmo DES, el algoritmo 3DES y el algoritmo 7DES. Estas soluciones S&D forman la jerarquía mostrada en la Fig. 4. Esta jerarquia está compuesta por la Clase S&D crypto y tres Patrones S&D pertenecientes a esta Clase S&D. Los Patrones S&D son: el DES, el 3DES y el 7DES. Se pueden ver a continuación en la Fig.4.

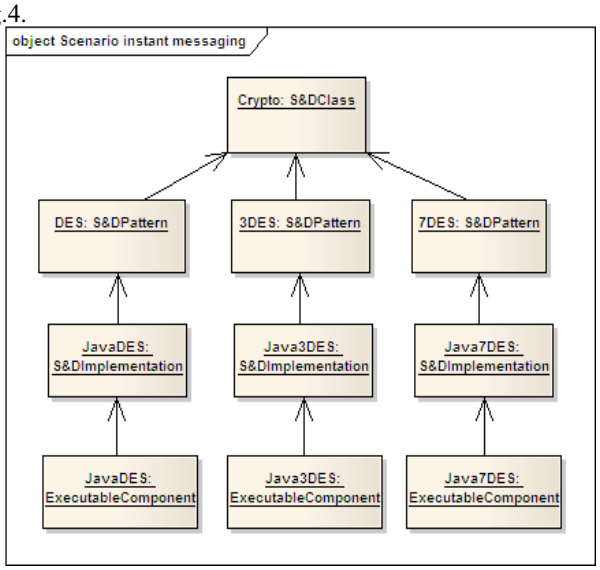


Figura 4.

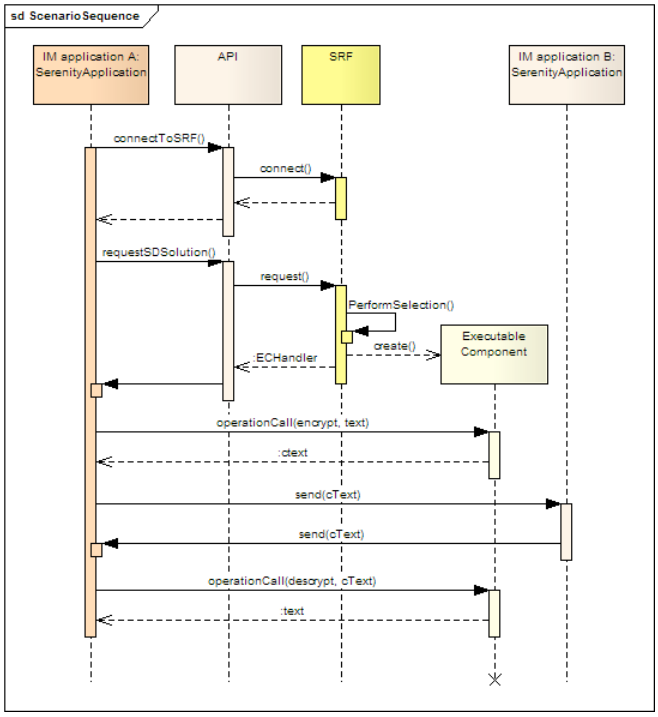
Además de los artefactos S&D, el escenario incluye la aplicación de mensajería instantánea (MI). La aplicación MI se comunica por medio de sockets TCP/IP. La aplicación MI

permite al usuario seleccionar el rol que va a desempeñar en la comunicación (servidor/cliente) y especificar la dirección IP para la conexión. La aplicación MI se muestra como una ventana (es una aplicación gráfica) donde el usuario puede escribir el texto que va a mandar y puede leer el texto que le manda la otra aplicación con la que se está comunicando. Hay una segunda ventana de aplicación que muestra tres botones diferentes, uno por cada método de criptografía posible. Usando esta segunda ventana el usuario puede simular un cambio de contexto.

B. *Funcionamiento del escenario*

Al principio ambas aplicaciones solicitan el mismo artefacto S&D al SRF (en este escenario ambas aplicaciones solicitan el Patron S&D DES).

Como se muestra en la Fig. 5, las aplicaciones MI empiezan creando una conexión con el SRF. Esto se realiza creando un objeto de la clase SRF_AP_AccessPoint. A continuación, la aplicación MI solicita una solución S&D. Para hacer esto, la aplicación crea un objeto de la clase SerenityExecutableComponent_AP. El constructor de este objeto incluye una petición al Componente Ejecutable que quiere usar la aplicación. Como se muestra en el diagrama el SRF procesa las solicitudes, instancia un Componente Ejecutable y devuelve un ECHandler a la aplicación MI. En este punto la aplicación MI usa el Componente Ejecutable para cifrar el texto. El texto cifrado (ctext) se envia a la otra aplicación MI que usa otro SRF. La aplicación MI que recibe el texto cifrado usa otra instancia del mismo Componente Ejecutable para cifrar/descifrar texto



VI. CONCLUSIONES

Este trabajo presenta un enfoque para desarrollar aplicaciones seguras usando soluciones de seguridad y dependabilidad intercambiables en tiempo de ejecución. Este enfoque, llamado SERENITY, se compone de dos frameworks.

Por un lado, el Framework de Desarrollo SERENITY (SDF) incluye conceptos, procesos y herramientas que ayudan al desarrollo de soluciones S&D y aplicaciones de seguridad. Estas soluciones S&D se implementan por medio de cuatro tipos distintos de artefactos S&D: (i) Clases S&D, (ii) Patrones S&D, (iii) Implementaciones S&D y (iv) Componentes Ejecutables.

Por otra parte, el framework de tiempo de Ejecución SERENITY (SRF) completa la arquitectura de software abierto. Este ofrece a las aplicaciones de seguridad las soluciones desarrolladas por el SDF. Es importante remarcar que el SRF incluye mecanismos de monitorización que garantizan que las soluciones S&D se están ejecutando correctamente.

Actualmente, contamos con una API totalmente operativa. Los siguientes pasos que se darán con esta API son (i) desarrollar la API en otros lenguajes de programación, (ii) crear un plugging para un Java IDE (actualmente trabajando en un plugging para Eclipse), y (iii) integrar el uso de la API que proponemos en un proceso de desarrollo para aplicaciones seguras, Esto último forma parte de la tesis doctoral presentada en [17].

VII. REFERENCIAS

[1] Serenity project. Funded by European Commission. Directorate General Information Society & Media. Unit D4 - ICT for Trust and Security, under grant IST-027587. http://www.serenity-project.org, 2006.

[2] Sara Ishikawa, Christopher Alexander, Murray Silverstein. "A pattern language: Towns, buildings, construction". (center for environmental structure series), 1977.

[3] Ralph Johnson, Erich Gamma, Richard Helm, John Vlissides. "Design patterns: elements of reusable object-oriented software". 1995.

[4] D. Tyree, D. M. Kienzle, M. C. Elder, J. Edwards-Hewitt. "Security patterns repository". 2006.

[5] Microsoft Press. "Web service security: Scenarios, patterns, and implementation guidance for web services enhancements (wse) 3.0". 2006.

[6] R. Nagappan, C. Steel, R. Lai. "Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management". Prentice Hall, 2005.

[7] D. Tyree, D. Kienzle, M. Elder, and J. Edwards-Hewitt. "Security patterns template and tutorial". 2002.

[8] M. Schumacher, U. Roedig. "Security engineering with patterns". 2001.

[9] S. Rajput M, N. Delessy-Gassant, E. B. Fernandez, M. Larrondo-Petrie. "Patterns for application firewalls". 2004.

[10] Sergio A. Velastin, Boghos A. Boghossian, Benny Ping, Lai Lo, Jie Sun, Maria Alicia Vicencio-silva. "Prismatica: Toward ambient intelligence in public transport environments".

[11] Alfonso Garate, Imanol Lucas, Fagor Electrodomesticos. "Ambient intelligent technologies for home automation and entertainment".

[12] Rita Cucchiara, Andrea Prati, and Reggio Emilia Italy."T park: Ambient intelligence for security in public parks".

[13] Paolo Bresciani, Loris Penserini, Paolo Busetta, and Tsvi Kuflik. "Agent patterns for ambient intelligence". In 23rd International Conference on Conceptual Modeling (ER2004), pages 8-12.

[14] Holger Schultheis. "Ipra enhancing the sensing abilities of ambient intelligence".

[15] Stefania Costantini, Luis Moniz Pereira, Francesca Toni. "Towards a model of evolving agents for ambient intelligence".

[16] A. Maña, C. Rudolph, G. Spanoudakis, V. Lotz, F. Massacci, M. Melideo, J. M. López-Cobo.Integrating Security and Software Engineering: Advances and Future Vision, chapter Security Engineering for Ambient Intelligence: A Manifesto. IDEA Group, 2006. ISBN 1-59904-148-0.

[17] Daniel Serrano, Antonio Maña, Athanasios-Dimitrios Sotirious. "Towards precise and certified security patterns". In Proceedings of 2nd International Workshop on Secure Systems methodologies using patterns (Spattern 2008), pages 287-291, Turin, Italy, September 2008. IEEE Computer Society. ISBN 978-0-7695-3299-8.

[18] http://www.java.com/.

Índice de autores

JITEL 2009

Universidad Politécnica de Cartagena

Asociación de Ingeniería Telemática

Grupo de Ingeniería Telemática (UPCT)

Escuela Técnica Superior de Ingeniería de Telecomunicación

ISBN: 978-84-96997-27-1