

EDITORIAL

Bienvenido querido lector a una nueva entrega de nuestra revista Buran, que alcanza ya con esta su edición nº21. Como podrás observar las personas que formamos esta rama de estudiantes del IEEE, continuamos con nuestro afán de divulgar y promocionar toda aquella información referente a las tecnologías de la información. Una tarea que aunque realizamos con mucho gusto y de forma totalmente desinteresada no resulta de fácil compaginación con las tareas académicas propias que nos ocupan en esta universidad. Así que me gustaría reflejar en estas líneas un especial agradecimiento a los miembros de la rama que para esta edición les ha resultado si más no un tanto atípica debido a la marcha al extranjero de la gran mayoría de miembros habituales en ella.

Como habrás podido comprobar esta edición la hemos dedicado en exclusiva al mundo de las comunicaciones móviles, un mundo bullicioso y de bastante actualidad. Debido a la gran afluencia de artículos recibidos sobre el tema hemos decidido realizar esta edición especial con carácter monotemático que esperamos sea de tu agrado.

Queremos felicitar a todos aquellos colaboradores que desinteresadamente envían sus artículos a esta revista los cuales facilitan y posibilitan la edición de la misma y pedir disculpas a todos aquellos que no hayan sido incluidos en este número por llegar a tiempo o por no coincidir con la temática, alentándoles a próximas ediciones.

Cuando Eduard, el antiguo coordinador, me propuso la coordinación de la revista pensé que debía aceptarlo como un deber para con la Rama. Sin embargo el paso del tiempo a lo largo de su construcción me ha demostrado todo lo contrario. Llegados a este punto en el que la revista está ya en el punto de venta, en tus manos, me siento muy orgulloso de haber colaborado en la elaboración de este ente llamado Buran y que ha significado el reflejo de mi breve paso por esta rama de estudiantes.

Hector Julian Bertomeu
Vicepresidente de la Rama
Coordinador de Buran

COMITÉ EJECUTIVO DE LA SECCIÓN ESPAÑOLA DEL IEEE

Presidente: Prof. José Antonio Delgado Penín
Vicepresidente: Prof. Manuel Sierra Pérez
Secretario: Prof. Eduardo Bertrán Albertí
Tesorero: Prof. Andrés de Santos y Lleó
Responsable de la actividad de "Membership development": Prof. Magdalena Salazar Palma
Responsable de relaciones con las Ramas de estudiantes: Ing. Juan Jesús Rodriguez Yubero

COORDINACIÓN BURAN Héctor Julian Bertomeu

RAMA DE ESTUDIANTES DE BARCELONA

Presidente: José Luis Hernández Sánchez
Vicepresidente: Héctor Julian Bertomeu
Secretario: Xavier Bielsa
Tesorero: Alfredo C. López
Resto de colaboradores: Raúl Cortés Delgado
Diseño Portada: Jordi Núñez

REVISIÓN

Carles Gómez Montenegro
José A. López Salcedo
Miguel Angel Sastre
Josep Paradells

AGRADECIMIENTOS
II. Dir. Juan A. Fernández Rubio,
Ángel Cardama, Jose A. Delgado-Peníñ, Jorge Luis
Sánchez-Ponz y a los puntos de distribución en la
UPC: Abacus, CPET, y Kiosk Campus Nord.

We would also like to thanks Ms. Laura Durrett
(IEEE Student Services Manager), and IEEE
International for their helpful support, encouragement
and financial funding for distributing Buran across
south american Region 9 IEEE Student Branches.

IMPRESIÓN RET, s.a.l. FOTOMECÁNICA Sistemes d'Edició DEPÓSITO LEGAL B-19.950-96

SCI UPC, 2004 (7338)

La organización se reserva el derecho de publicar los artículos. La opinión expresada en los artículos no tiene por qué coincidir con la de la organización.

Agradecemos las colaboraciones hechas desinteresadamente, y a causa de la falta de espacio, pedimos disculpas a todas aquellas personas a las cuales no se les ha publicado su colaboración. Esperamos que en un próximo número tengan cabida.

MONTAR UN PUNTO DE ACCESO INALÁMBRICO 802.11 EN LINUX

Juan Hernández-Serrano, Josep Pegueroles

jserrano@entel.upc.es, josep.pegueroles@entel.upc.es

Resumen - La gente se mueve, las redes no. Estas dos sentencias definen más que nada la explosión de las redes inalámbricas de área local o WLAN (Wireless Local Area Network). En unos pocos años el precio de los dispositivos WLAN ha bajado lo suficiente como para que se plantee su adquisición para usos domésticos. Aun así el precio del punto de acceso o AP (Access Point) sigue siendo elevado, y en la mayoría de casos su capacidad sobrepasa los pequeños requerimientos de una red inalámbrica doméstica. En este artículo explicamos como montar un AP inalámbrico con un ordenador, una tarjeta WLAN y software de código libre.

1 INTRODUCCIÓN

La gente se mueve, las redes no [MG02]. Con estas dos sentencias queda más que definida la explosión de las redes inalámbricas de área local o WLAN (*Wireless Local Area Network*).

Con el paso de los años el mundo es cada vez más móvil. Especialmente en el mundo laboral, la facilidad para dar cobertura a nuevas sedes y espacios, la flexibilidad para la conectividad de nuevos trabajadores, y especialmente el ahorro en infraestructura de cableado; han impulsado la implantación de WLANs.

Actualmente estamos viviendo la explosión del mundo inalámbrico. La telefonía móvil ha basado su éxito en permitir que la gente se comunique entre sí independiente de su localización, y parece que el acceso a redes de datos (p.e. Internet) siga el mismo camino. La tecnología de redes inalámbricas que más ha fructificado es 802.11, que pone la base para crear WLANs.

En unos pocos años el precio de los dispositivos WLAN ha bajado lo suficiente como para que se plantea su adquisición para usos domésticos y no sólo con carácter empresarial. Aun así el precio del punto de acceso o AP (*Access Point*) sigue siendo elevado, y en la mayoría de casos su capacidad sobrepasa los pequeños requerimientos de una red inalámbrica doméstica con más de 5 usuarios simultáneos.

En este artículo explicamos como montar un AP inalámbrico con un ordenador, una tarjeta wireless y software de código libre. En la sección 2 introducimos la tecnología de red WLAN. A continuación se introduce el concepto de

puente de red o *bridge*. La sección 4 indica los requerimientos de hardware y software para implementar un AP casero funcionando como *bridge* entre una red 802.11 y una red ethernet (802.3). A continuación se comentan los pasos de configuración para la puesta a punto del AP y puente red. Y finalmente concluimos con un breve estudio de la situación actual de las WLAN y de su futuro.

2 REDES INALÁMBRICAS DE ÁREA LOCAL

La WLAN es un tipo de LAN en la que la comunicación entre nodos se transporta a través de un medio inalámbrico, normalmente radio-frecuencia, en vez de a través de cables. Este tipo de comunicación permite combinar la conectividad de los nodos con la movilidad de los mismos.

Las redes WLAN deben proporcionar al menos el mismo nivel de funcionalidad que las redes LAN, con la ventaja adicional de tener menores restricciones físicas y sobre todo de la movilidad tanto de los usuarios como de la propia red.

Si bien se han desarrollado diversos estándares para WLANs, los dos principales son la americana IEEE 802.11 y la europea HyperLAN, aunque actualmente se puede considerar que la gran mayoría de dispositivos WLAN están basados en la familia de especificaciones IEEE 802.11. Veamos pues como funciona.

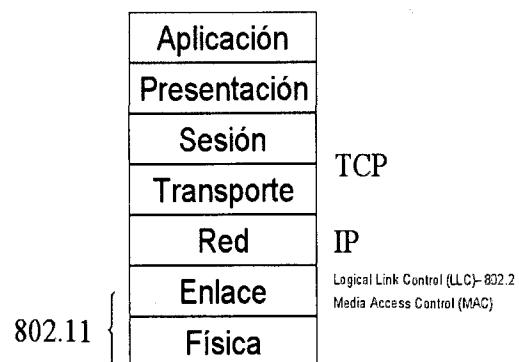


Figura 1. 802.11 en la pila OSI

2.1 Estándar IEEE 802.11 para WLANs

802.11 es una familia de especificaciones desarrollada por el IEEE para tecnología WLAN. 802.11 especifica una interfaz «a través del aire» entre un cliente inalámbrico y una estación base, o entre varios clientes inalámbricos. Su objetivo principal es el de simular el comportamiento de redes ethernet (802.3) usando radio-frecuencia en vez de cables.

Tal y como se muestra en la Figura 1, 802.11 define dos capas lógicas de la pila OSI (Open System Interconnection):

- Control de Acceso al Medio o MAC (Medium Access Control)
- Física o PHY (Physical)

802.11 se define originalmente con una tasa de transferencia de hasta 2 Mb en la banda de 2,4 GHz y utilizando modulaciones de espectro expandido (FHSS - Frequency-Hopping Spread Spectrum y DSSS Direct-Sequence Spread Spectrum), aunque actualmente se define sobre las siguientes 3 capas físicas (véase Tabla 1):

- 802.11a. Opera en la banda de 5 GHz y provee una tasa de transmisión de hasta 54 Mbps a una distancia máxima de unos 45 metros (150 pies). Utiliza técnicas de codificación OFDM (*Orthogonal Frequency Division Multiplexing*). Tiene peor comportamiento frente a interferencias de radio-frecuencia que 802.11b, pero sin embargo se comporta mejor para el transporte de voz y datos multimedia así como en escenarios con una alta densidad de usuarios.

- 802.11b (también denominada 802.11 High Rate). Opera en la banda de 2,4 GHz y provee una tasa de transmisión de hasta 11 Mbps a una distancia máxima de unos 75 metros (250 pies). Utiliza sólo DSSS debido a que DSSS es capaz de manejar mejor que FHSS (también recomendado en la norma original 802.11) las señales de baja intensidad. Con DSSS pueden extraerse los datos de un fondo de interferencias sin necesidad de retransmitirlos. 802.11b es el primer estándar que provee a las redes WLAN de una funcionalidad comparable a la de las redes LAN (ethernet familia 802.3). Su ventaja principal respecto a 802.11a es que requiere un menor número de puntos de acceso para cubrir grandes superficies de terreno. Además es el primer estándar en cubrir el modo de funcionamiento Ad-Hoc.

- 802.11g. Opera en la banda de 2,4 GHz y provee tasas de transmisión de 6 a 54 Mbps. Al igual que 802.11b, utiliza 3 canales no-superpuestos, y como en 802.11a, usa modulación OFDM, aunque para garantizar la compatibilidad «hacia atrás» con 802.11b, 802.11g usa también modulación CCK (*Complementary Code Keying*) y opcionalmente PBCC (*Packet Binary Convolutional Coding*). Esta compatibilidad «hacia

atrás» viene a decir que cuando un dispositivo móvil 802.11b se adhiere a un punto de acceso 802.11g, todas las conexiones de este punto de acceso se reducen a velocidades 802.11b. 802.11b se adhiere a un punto de acceso 802.11g, todas las conexiones de este punto de acceso se reducen a velocidades 802.11b.

Además se definen:

- 802.11e. Provee Calidad de Servicio (QoS - *Quality of Service*) para aplicaciones WLAN, como por ejemplo Voz sobre IP inalámbrica (VoWIP - *Voice over Wireless IP*).

- 802.11h. Se trata de un suplemento a la capa MAC para que cumpla las regulaciones europeas de la banda de 5GHz, que obligan a utilizar tanto control de potencia de transmisión (TPC - *Transmission Power Control*) como selección dinámica de frecuencias (DFS - *Dynamic Frequency Selection*). TPC limita la potencia de transmisión a la mínima necesitada para llegar hasta el usuario más lejano. DFS selecciona el canal de radio al punto de acceso de manera que minimice las interferencias con otros sistemas, en concreto con el radar.

- 802.11i. Borrador de estándar cuyo objetivo es incrementar la seguridad 802.11. Describe la transmisión de datos cifrados tanto en 802.11a como en 802.11b. Actualmente se usa WPA (*Wi-Fi Protected Access*) como solución interina hasta que se apruebe 802.11i como estándar, por lo que por el momento las mejoras en seguridad que provee son similares a las de WPA.

Variedades	Velocidad	Frecuencia	Distancia
802.11a	< 54 Mb	5 GHz	< 45 m
802.11b	5,5 y 11 Mb	2,4 GHz	< 75 m
802.11g	< 54 Mb	2,4 (y 5) GHz	< 75 m

Tabla1. Variedades según capa física de 802.11

2.2 Topologías 802.11

El bloque básico de construcción de una red 802.11 es el BSS (*Basic Service Set*), que es simplemente una agrupación de estaciones WLAN que se comunican entre sí. La comunicación entre estaciones se establece dentro del área de servicio básico o BSA (*Basic Service Area*).

Los BSSs son útiles para dar cobertura a pequeñas oficinas y casas, pero no sirven para dar cobertura a grandes áreas. Para poder dar una cobertura mayor, 802.11 permite unir o enlazar varios BSSs en un ESS (*Extended Service Set*). 802.11 no especifica ningún tipo de infraestructura, sino sólo qué servicios o *service set* se ha de proveer con ella.

Independientemente del tipo de *service set* utilizado, se definen también dos tipos de topología de red para WLAN:



- **Modo independiente o ad-hoc.** Los terminales se conectan unos a otros de igual a igual para establecer una red dinámica en la que todos los nodos o terminales tienen las mismas funciones de enrutamiento

- **Modo infraestructura.** Existe un punto de infraestructura fijo denominado punto de acceso (AP) al que se conectan los terminales. La conexión entre terminales se produce, por lo tanto, a través de dicho AP, que además suele ejercer funciones de puente entre dos redes, p.e. extender la red ethernet de la oficina a los dispositivos WLAN.

El escenario más común es el de una red WLAN 802.11 trabajando en modo infraestructura para extender la red local o LAN (*Local Area Network*), basada generalmente en tecnología ethernet (802.3), a los nuevos dispositivos inalámbricos. En este tipo de escenarios el AP suele funcionar como puente o *bridge* entre las dos redes físicas, con lo que a efectos del usuario inalámbrico su dispositivo se encuentra directamente conectado a la red ethernet.

En este artículo explicamos como montar un AP 802.11 funcionando como *bridge* entre las red 802.11 y una red ethernet.

3 EL PUENTE DE RED O BRIDGE

Un puente de red es básicamente un nexo de unión entre dos o más segmentos de red. Como la unión se produce al nivel de capa 2 (enlace), todos los protocolos pueden correr de forma transparente sobre él. Para entender como funciona un puente de red, es por lo tanto importante entender qué es eso del segmento de red.

Un segmento de red es una sección de medios de red que conecta dispositivos. Por ejemplo, supongamos que tenemos 3 ordenadores A, B y C. El ordenador A tiene 2 tarjetas de red, y los ordenadores B y C sólo una. Un cable ethernet que une A y B creará un segmento de red S_{AB} , y otro que une A y C creará otro segmento S_{AC} . Véase Figura 2. Si quisieramos simular una conexión directa entre B y C, es decir; simular que están conectados entre ellos por un sólo cable ethernet (un sólo segmento de red) deberíamos configurar el ordenador A como puente de red que une los dos segmentos de red S_{AB} y S_{AC} para establecer un sólo segmento entre B y C S_{BC} . Véase Figura 3.

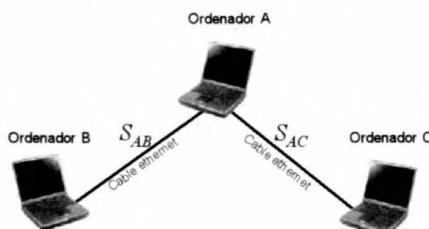


Figura 2. Segmentos de red

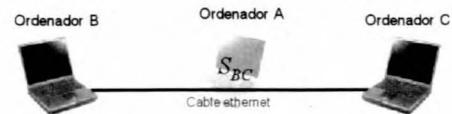


Figura 3. Ordenador A actuando como puente de red

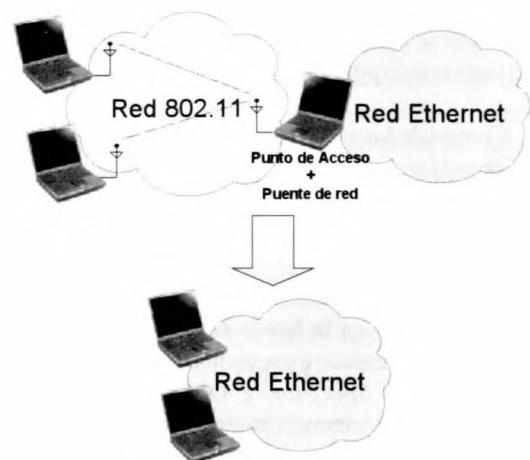


Figura 4. Punto de Acceso 802.11 y puente de red

Nótese que al configurarse el ordenador A como puente de red, A es ahora transparente entre B y C y por lo tanto invisible. Además, nótese que el puente de red lo que simula es una conexión a nivel de enlace (MAC), en este caso una conexión ethernet, muy diferente a la conexión IP (independiente de la red física) que lograríamos con un enrutador IP o *router*.

En el escenario de este artículo vamos a configurar un AP con *bridge* entre una red WLAN 802.11 y una red ethernet. En este caso, podríamos entender, simplificando, que el AP extiende la red ethernet a los dispositivos inalámbricos de forma que estos dispositivos parezcan conectados directamente a la red física ethernet. De esta forma nuestro escenario queda definido como en la Figura 4.

4 REQUERIMIENTOS DE HARDWARE Y SOFTWARE

El objetivo es montar un AP 802.11 que funcione como puente de red con una red ethernet 802.3. La elección de hardware y software está estrechamente ligada. La elección de un determinado software determinará el hardware a comprar y viceversa.

4.1 Hardware necesario

Sólo necesitamos un ordenador, una tarjeta de red 802.11 y una tarjeta de red ethernet.

4.1.1 Un ordenador

El AP se ha probado con éxito en un ordenador PII 366Mhz con 128Mb de RAM y debería funcionar incluso con máquinas menos potentes. Es importante que exista una distribución de Linux reciente que sea compatible con nuestro ordenador, por lo que recomendamos un ordenador con procesador Pentium X o AMD. Además es importante que el ordenador tenga soporte de PCI (ordenador) o de PCMCIA(portátil), aunque raro será el caso (sino prácticamente imposible) en el que no dispongamos de ellos.

4.1.2 Tarjeta de red 802.11

La elección de la tarjeta es crítica y determinará el éxito o el fracaso de nuestro propósito. Es fundamental que la tarjeta adquirida acepte el modo de funcionamiento de Host AP (también llamado modo Master), y sobretodo que tenga soporte de drivers para Linux. Además, hemos de tener en cuenta que tipo de variedad 802.11 deseamos implementar (802.11a, 802.11b u 802.11g). Descartada la tecnología 802.11a por obsoleta y por su predisposición única al ámbito empresarial, hemos de elegir entre la tecnología 802.11b y la 802.11g.

Lo más sencillo es optar por la tecnología 802.11b, que al ser más antigua, tiene un mayor soporte; y buscar tarjetas que soporten el software Host AP que es, desde nuestro punto de vista, la mejor implementación software de todas las funciones de un AP, incluidas muchas de las nuevas funciones criptográficas para privacidad y autenticación (Wi-Fi Protected Access - WPA). El software Host AP puede usarse con tarjetas basadas en los chipsets Intersil Prism 2, 2.5 y 3 que las proveen varios fabricantes (D-Link, LinkSys, Netgear, SMC ...). Aún así hay muchas tarjetas que inicialmente pone que están basadas en un determinado chipset, luego resulta que no es cierto y dejan de ser útiles. Recomendamos encarecidamente a todo el que quiera adquirir un tarjeta con este chipset que lea primero esta la URL en [4] a fin de asegurarse la adquisición de una tarjeta plenamente compatible con Linux y con soporte de modo Host AP.

La otra opción es optar por la tecnología 802.11g, más moderna y por lo tanto con un soporte menor para Linux. 802.11g ofrece sin embargo algunas ventajas. Además, claro está, del incremento de velocidad (54Mbs frente a 11Mbps), 802.11g es la tecnología que se está actualmente imponiendo, ya que no es más que la evolución natural de la tecnología 802.11b. Pronto toda la funcionalidad de 802.11b sobre Linux será extendida a 802.11g, sobre la que ya se soportan las funciones más básicas. Las tarjetas 802.11g adquiridas deben estar basadas en los chipsets Intersil Prism GT o Duette y ser soportadas por el driver prism54 (véase una lista de tarjetas soportadas en [5]). En nuestro caso hemos optado por la tecnología 802.11g y hemos adquirido la tarjeta PCMCIA SMC2835W con un resultado de éxito.

4.1.3 Tarjeta de red ethernet

Cualquier tarjeta debería funcionar. De todas formas sería bueno asegurarse de que la tarjeta tiene desarrollados drivers para Linux, aunque como ya hemos dicho es muy raro encontrar una tarjeta ethernet no soportada.

4.2 Software necesario

Tan importante como el hardware adquirido es el software que vamos a utilizar para montar tanto el AP como el puente de red.

4.2.1 Punto de Acceso (AP)

La mayoría de funciones del AP se realizan directamente a nivel de *driver* o de *firmware*. El *firmware* podría considerarse como un «mini» sistema operativo que utiliza cada dispositivo para implementar diferentes funciones; y el *driver*, como un interfaz de comunicación entre nuestro sistema operativo y el *firmware* del dispositivo.

Si hemos optado por la tecnología 802.11b, habremos adquirido una tarjeta con chipset Intersil Prism 2, 2.5 ó 3 y el driver que hemos de usar es Host AP ([6]). Este driver se distribuye por defecto con cualquier distribución reciente de Linux (SuSE, Fedora, Debian...) y por lo tanto la tarjeta debería funcionar con tan sólo insertarla. De todas formas para poder acceder a las nuevas funciones de seguridad implementadas seguramente tengamos que actualizarnos a la última versión del driver.

En caso de haber optado por la tecnología 802.11g, habremos adquirido una tarjeta con chipset Intersil Prism GT o Duette y el driver a usar es prism54 ([7]). Este driver todavía no está integrado en las distribuciones de Linux con lo que tendremos que compilarlo nosotros mismos. Para poder compilar el driver necesitamos una versión del *kernel* superior a la 2.4.23, el código fuente de dicho *kernel*, la herramientas de compilación (gcc, make, ...), el código fuente del driver ([8]) y el último *firmware* ([9]). Los pasos de compilación los tenemos en el mismo sitio web.

Una vez que tenemos el driver funcionando hemos de obtener los paquetes *wireless-tools* y *wireless-extensions*. El paquete *wireless-extensions* viene incluido como parte del *kernel*, con lo que a la práctica sólo hemos de obtener las *wireless-tools*. Normalmente tendremos este paquete precompilado dentro de nuestra distribución (si es actual), pero si no podemos descargarlo de [10].

El paquete *wireless-extensions* es una API genérica que permite al driver mostrar al usuario información sobre configuración y estadísticas comunes a las WLANs. Su belleza reside en que este paquete puede soportar todas las variaciones de WLANs independientemente de su tipo, y además los parámetros pueden cambiarse "al vuelo" sin necesidad de reiniciar el driver.



El paquete *wireless-tools* provee un paquete de herramientas para manipular las *wireless-extensions*. Utilizan un interfaz modo texto bastante tosco, pero que nos permite configurar y mostrar de forma más inteligible todas las *wireless-extensions*. De hecho, hay otras muchas herramientas para manipular las *wireless-extensions*, pero *wireless-tools* es siempre la implementación de referencia. Las 4 herramientas que provee este paquete son:

- *iwconfig* para manipular los parámetros básicos de la conexión inalámbrica
- *iwlist* nos permite iniciar el escaneo del medio y listar frecuencias, bit-rates, claves de encriptación...
- *iwspy* permite obtener la calidad del enlace para cada nodo
- *iwpriv* permite manipular parámetros específicos de cada driver / firmware (privados)

4.2.2 Puente de red

Para montar el puente de red sólo necesitamos el paquete *bridge-utils*, que viene incluido en la mayoría de distribuciones de Linux, y tener las dos tarjetas de red (la ethernet y la 802.11) funcionando.

Normalmente el código de las *bridge-utils* se compila como un módulo dentro del sistema operativo. Si el módulo está correctamente configurado e instalado, se cargará automáticamente la primera vez que llamemos al comando *brctl*. Si todo ha ido bien entonces al introducir el comando *brctl* debe aparecernos una pequeña sinopsis del comando.

```
# brctl
commands:
addbr      <bridge>          add bridge
addif      <bridge> <device>    add interface to bridge
delbr      <bridge>          delete bridge
delif      <bridge> <device>    delete interface from bridge
show
showmacs   <bridge>          show a list of bridges
showstp    <bridge>          show a list of mac addrs
showageing <bridge> <time>    show bridge stp info
setageing  <bridge> <time>    set ageing time
setbridgepriorio <bridge> <prio>  set bridge priority
setfd      <bridge> <time>    set bridge forward delay
setgcint   <bridge> <time>    set garbage collection interval
sethello   <bridge> <time>    set hello time
setmaxage  <bridge> <time>    set max message age
setpathcost <bridge> <port> <cost>  set path cost
setportpriorio <bridge> <port> <prio>  set port priority
stp       <bridge> <state>   turn stp on/off
```

5 CONFIGURACIÓN DEL AP Y EL PUENTE DE RED

Si el módulo del driver de la tarjeta WLAN 802.11 está cargado al ejecutar el comando *iwconfig* deberíamos ver algo como:

```
# iwconfig
lo      no wireless extensions.

eth0    no wireless extensions.

eth1    NOT READY!  ESSID:off/any
        Mode:Managed Channel:6 Access Point: 00:0C:00:00:00:00
        Tx-Power=31 dBm  Sensitivity=-0/200
        Retry min limit:0 RTS thr=0 B Fragment thr=0 B
        Encryption key:off
        Link Quality:0 Signal level:0 Noise level:0
        Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
        Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

En lo que observamos que la tarjeta 802.11 está asociada al dispositivo eth1. En caso de no ver ningún dato, puede que se deba a que no tenemos cargado el módulo de la tarjeta. Podemos cargarlo con *modprobe prism54* si se trata de una tarjeta 802.11g o con *modprobe* y el nombre del módulo en caso de trabajar con otra tarjeta.

Una vez cargado el módulo de la tarjeta de red correctamente debemos asignar los parámetros básicos de nuestra conexión inalámbrica. Estos son:

- **essid**. identifica el área en el que nuestro AP va a dar cobertura. El AP emitirá constantemente este parámetro por el medio. Cuando un cliente se quiera conectar a nuestro AP buscará el essid de nuestro AP en el medio y entonces le localiza físicamente.

- **channel**. Se refiere al canal de comunicaciones o frecuencia a la que queremos transmitir. Los canales se establecen con saltos de 5 Mhz de tal forma que el canal 1 está en 2,412 GHz, el 2 en 2,417 y así hasta el 14. Para cubrir la banda de 5 GHz (sólo 802.11a y g) también se establecen otros 11 canales separados 10 MHz (ó 20 MHz) empezando en 5,17 GHz. Debemos procurar configurar algún canal diferente al de otras posibles WLAN del entorno.

- **mode**. denota el modo de trabajo de la tarjeta. En principio hay 4:

- mode **master**: trabaja como un AP

- mode **managed** (modo por defecto): trabaja como una estación cliente que se conectará al AP definido en essid o a cualquiera si no se define dicho parámetro

- mode **ad-hoc**: para funcionar como una red ad-hoc en que todas las estaciones se tratan de igual a igual (sin infraestructura)

- mode **monitor**: permite escuchar el medio de forma pasiva, es decir, sin asociarse o identificarse ante ningún otro dispositivo.

Es decir, que si queremos crear un AP con essid CASABLANCA en el canal 11 debemos introducir

```
# iwconfig eth1 mode master essid CASABLANCA channel 11
```

y si ahora tecleamos *iwconfig* deberá aparecernos algo como lo siguiente

```
# iwconfig
lo      no wireless extensions.

eth0    no wireless extensions.

eth1    NOT READY!  ESSID:"CASABLANCA"
        Mode:Master  Channel:11  Access Point: 00:00:00:00:00:00
        Tx-Power=31 dBm  Sensitivity=-200
        Retry min limit:0  RTS thr=0 B  Fragment thr=0 B
        Encryption key:off
        Link Quality:0  Signal level:0  Noise level:0
        Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
        Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

donde se puede apreciar que todavía vemos la tarjeta como NOT READY!. Esto es debido a que debemos darla de alta como tarjeta de red en el sistema operativo, lo que se hace con el comando `ifconfig eth1 up`. Si ahora volvemos a ejecutar `iwconfig` obtenemos:

```
# iwconfig
lo      no wireless extensions.

eth0    no wireless extensions.

eth1    IEEE 802.11b/g  ESSID:"CASABLANCA"
        Mode:Master  Channel:11  Access Point: 00:04:E2:A4:B2:8A
        Bit Rate:54Mb/s  Tx-Power=31 dBm  Sensitivity=-20/200
        Retry min limit:8  RTS thr:2347 B  Fragment thr:2346 B
        Encryption key:off
        Link Quality:0  Signal level:0  Noise level:0
        Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
        Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

Como se puede observar ahora si que tenemos nuestro AP activo. Está utilizando el protocolo 802.11b y g; está dado de alta en el essid CASABLANCA; trabaja como AP (mode: Master); funciona sobre el canal 11; y tiene como dirección física única 00:04:E2:A4:B2:8A. De momento, como se puede observar, no hay clave de encriptación.

En este punto tendríamos un AP funcionando y anunciándose en el medio, pero que todavía no da acceso a nada. Como nuestro objetivo es dar acceso a la red ethernet (en el dispositivo eth0), debemos configurar el puente de red entre la tarjeta 802.11 (eth1) y la ethernet (eth0).

El primer paso será activar ambas tarjetas en el sistema pero sólo a nivel físico (deshabilitando TCP/IP), lo que realizamos con:

```
# ifconfig eth0 0.0.0.0 up
# ifconfig eth1 0.0.0.0 up
```

Una vez activadas creamos un puente de red que llamaremos *wlan-bridge* con:

```
# brctl addbr wlan-bridge
```

Ahora añadimos las dos tarjetas al puente de red con:

```
# brctl addif wlan-bridge eth0
# brctl addif wlan-bridge eth1
```

Y para terminar activamos el puente de red como un dispositivo de red en el sistema con:

```
# ifconfig wlan-bridge up
```

En este punto cualquier usuario 802.11 que se conecte a nuestro AP tendrá acceso transparente a la red ethernet, luego ya hemos alcanzado nuestro objetivo inicial de dar acceso a la red ethernet a los estaciones de la WLAN.

Ahora bien, muy posiblemente no deseemos que cualquier estación 802.11 pueda conectarse a nuestra red ethernet (posiblemente conectada a Internet). Imaginemos que montamos el AP en nuestra casa y lo que hacemos es dar Internet gratuito a nuestros vecinos. Seguramente, si los que pagamos somos nosotros sólo querremos que se conecten nuestros ordenadores, es decir, querremos un mecanismo de autenticación/autorización que deniegue el acceso al resto.

Todas las tarjetas de red 802.11 con chipset Intersil Prism X (como la nuestra) permiten al menos dos tipos de autenticación:

- Autenticación basada en secreto compartido (parte del WEP - *Wired Equivalent Privacy*)

- Autenticación basada en dirección MAC única de cada dispositivo de red

La primera es estándar de cualquier dispositivo 802.11 y se basa en la compartición de un secreto entre el AP y las estaciones cliente. Este secreto debe ser de 40 bits ó 104 bits, y se usa para cifrar/descifrar la información enviada entre el AP y las estaciones. Para configurar una clave en el AP debemos escribirla en hexadecimal (10 dígitos - 40 bits, 26 dígitos - 104 bits) o en ASCII (5 caracteres - 40 bits, 13 caracteres - 104 bits). Por ejemplo, si quisieramos establecer la contraseña en código ASCII *qwert* (7177657274 en hexadecimal) la podemos introducir como:

```
# iwconfig eth1 enc s:"qwert"
```

O

```
# iwconfig eth1 enc 7177657274
```

Evidentemente se ha de configurar la misma clave en cada estación que vaya a conectarse al AP, y sobretodo utilizar claves seguras y actualizarlas con frecuencia. Existen diversas páginas de Internet desde las que podemos generar claves seguras para WEP, un ejemplo lo podemos encontrar en [11].

La segunda forma de autenticación no es estándar del protocolo 802.11, pero sí que viene implementada en la mayoría de dispositivos. Se trata simplemente de establecer una política de aceptación basada en la dirección MAC única de cada tarjeta de red. Al no tratarse de un estándar la debemos configurar con el comando `iwpriv` que nos permite manipular parámetros específicos de cada driver. Lo primero es establecer que política vamos a usar de las tres disponibles:

- `MAC_POLICY_OPEN = 0` : Se acepta a cualquier cliente.

- `MAC_POLICY_ACCEPT = 1` : Se acepta a cualquier cliente excepto a aquellos cuya MAC esté en la lista.



-MAC_POLICY_REJECT=2 : Se rechaza a cualquier cliente excepto a aquellos cuya MAC esté en la lista.

Y después añadir las MACs de las tarjetas de red que no queremos dejar acceder o a las que solamente queremos dejar acceder. Por ejemplo, si en nuestra red sólo tenemos un ordenador que se va conectar vía 802.11, y cuya dirección MAC única de su tarjeta de red es 00:04:E2:A5:C3:DE, deberíamos ejecutar los siguientes comandos:

```
# iwpriv eth1 setPolicy 2  
# iwpriv eth1 addMac 00:04:E2:A5:C3:DE
```

Con lo que nos aseguraríamos que sólo se utiliza nuestra red desde esta máquina.

Nótese que ambos sistemas de autenticación son perfectamente compatibles y pueden complementarse.

6 SUMARIO Y CONCLUSIÓN

Actualmente las redes inalámbricas están teniendo un desarrollo muy grande tanto a nivel corporativo como doméstico, sin embargo las soluciones domésticas suelen tener todavía un precio elevado, especialmente cuando hablamos del AP. En este artículo hemos explicado como montar un AP en un ordenador con Linux ya conectado a una red ethernet y una tarjeta de red 802.11. También hemos explicado como implementar unos mecanismos de seguridad básicos pero que, por otra parte, son perfectamente válidos para evitar ataques «casuales».

De todas formas los mecanismos de seguridad que actualmente vienen por defecto presentan muchísimas debilidades y se han presentado diversas formas de saltárselas en un periodo de tiempo muy reducido. Airsnort ([12]) es un ejemplo de herramienta de código libre que circula por Internet que nos permite recuperar la clave WEP de cifrado/descifrado en muy pocas horas. Por ello los fabricantes han implementado WPA (*Wi-Fi Protected Access*) que mejora sustancialmente el estándar actual, y que está previsto que salga como estándar de forma inminente bajo el nombre de 802.11i. Algunas de las tarjetas que podemos adquirir soportan WPA, pero en general los drivers de Linux no son los suficientemente funcionales en este aspecto (por el momento).

7 AGRADECIMIENTOS

Este trabajo ha sido soportado por el proyecto DISQET [CICYTTIC2002-00249], dentro del Plan Nacional de I+D.

REFERENCIAS

- [1] Gast, M. *802.11 Wireless Networks – The Definitive Guide*. O'Reilly, Abril 2002. ISBN 0-596-00183-5
- [2] Nichols, R.K.; Lekkas, P.C. *Seguridad para comunicaciones inalámbricas*. McGraw-Hill, 2003. ISBN 84-481-3782-5
- [3] Geier, J. *Wireless LANs. Implementing High Performance IEEE.11 Networks*. SAMS. 2001.

REFERENCIAS A HIPERTEXTO

- [4] http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Linux.Wireless.drivers.802.11b.html#Prism2
- [5] http://prism54.org/supported_cards.php
- [6] <http://hostap.epitest.fi/>
- [7] <http://prism54.org/>
- [8] <http://prism54.org/download/>
- [9] <http://prism54.org/~mcgroat/firmware/>
- [10] http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html
- [11] <http://www.warewolflabs.com/portfolio/programming/wepskg/wepskg.html>
- [12] <http://airsnort.shmoo.com/>

AUTORES



Josep Pegueroles (Tortosa 1974) recibió el título de Ingeniero de Telecomunicación en 1999 y el de doctor en 2003, ambos por la UPC. En 1999 entró a formar parte del Grupo de Seguridad de la Información - ISG dentro de la Línea de Investigación de Servicios Telemáticos - SERTEL del Departamento de Ingeniería Telemática (<http://sertel.upc.es>). Actualmente es profesor asociado de la ETSETB y sus intereses de investigación incluyen la seguridad para servicios multimedia en red y las comunicaciones seguras de grupo.



Juan Hernández (Salamanca 1979) recibió el título de Ingeniero de Telecomunicaciones por la UPC en 2002. Ese mismo año pasó a formar parte del Grupo de Seguridad de la Información - ISG dentro de la Línea de Investigación de Servicios Telemáticos en el Departamento de Ingeniería Telemática de la UPC. Actualmente es estudiante de doctorado de la Universidad Politécnica de Cataluña (UPC) en el Departamento de Ingeniería Telemática y su investigación se centra en seguridad en redes ubicuas y gestión de claves de grupo para redes multicast.



HACIA UNA WEB INDEPENDIENTE DEL DISPOSITIVO MEDIANTE CC/PP

José Luis Ferrer Riera, Anna Calveras Augé

Departamento de Ingeniería Telemática, UPC

Grupo de Redes Inalámbricas

jfer6322@alu-etsetb.upc.es, anna.calveras@mat.upc.es

Resumen- Composite Capabilities/Preferences

Profiles (CC/PP) es el nuevo lenguaje estándar creado por el World Wide Web Consortium (W3C) para que la gran diversidad de dispositivos que disponen, o dispondrán, de un acceso a Internet (móvil, PDA, PC, TV...), sean capaces de expresar sus capacidades y las preferencias del usuario mediante perfiles, tal y como su nombre indica. Para expresar estas características, los perfiles CC/PP emplean Resource Descripción Framework (RDF), otro estándar del W3C y pilar de la web semántica, el cual, a su vez, se encuentra construido sobre Extensible Markup Language (XML).

La especificación User Agent Profile (UAProf) definida por la Open Mobile Alliance (OMA), antiguo WAPForum, utiliza CC/PP para la descripción de los teléfonos móviles. Se puede entender como el primer gran desarrollo CC/PP y ya se encuentra incluido en los últimos dispositivos móviles, implicando la existencia actual de millones de dispositivos que usan CC/PP.

Mediante CC/PP los dispositivos tienen la habilidad de enviar la información CC/PP conjuntamente con las peticiones HTTP a los servidores, de manera que éstos puedan procesar la información CC/PP y realizar la adaptación o selección de contenidos adecuados a las características del dispositivo, consiguiendo así una Web independiente del dispositivo, acercándose cada vez más a uno de las principales metas del W3C: El Acceso Universal a la Web.

En este artículo se pone de manifiesto la problemática actual de la negociación de contenidos utilizando HTTP/1.1 y se describe la especificación CC/PP, el estándar propuesto por W3C para solucionar esta carencia, justificando sus claves de diseño. También se describe el estándar de la OMA, UAProf, la primera implementación de CC/PP para la descripción de los terminales móviles que se encuentra incluido en la nueva especificación WAP 2.0.

Palabras clave- CC/PP, UAProf, RDF, Adaptación de contenidos, móvil, Web

¿POR QUÉ CC/PP?

Con la puesta en funcionamiento de los sistemas móviles de tercera generación (UMTS) y la gran utilización de redes basadas en el estándar IEEE 802.11 (Wireless LAN), se consigue una mayor capacidad en los enlaces inalámbricos, permitiendo la existencia actual de una gran heterogeneidad de dispositivos, como son los teléfonos móviles o las PDAs, capaces de acceder a multitud de contenidos web antes impensables: páginas con contenidos multimedia (audio, imágenes y video) o páginas con contenidos script, p.e. Javascript.

Actualmente, la especificación HTTP/1.1 (Hypertext Transfer Protocol)[1] permite la negociación de contenidos basada en servidor, donde la selección de la mejor representación para la respuesta HTTP es realizada por el servidor.

La selección se basa en las diferentes representaciones de la respuesta y de los contenidos de los siguientes campos de la cabecera de petición:

-Accept. El campo Accept se utiliza para especificar qué tipos MIME (Multipurpose Internet Mail Extensions) son aceptados con la respuesta.

-Accept-Charset. Este campo se puede usar para indicar qué tipos de caracteres se aceptan en la respuesta.

-Accept-Encoding. Este campo restringe la codificación de los contenidos que se aceptan en la respuesta.

-Accept-Language. Utilizado para restringir el grupo de lenguajes naturales que son preferidos como respuesta a una petición.

-User-Agent. Contiene la información que identifica al cliente que inicia la petición (user agent).



Un posible ejemplo de cabecera de petición sería:

```
User-Agent: Mozilla/4.04 (X11; I; SunOs 5.4 sun4m)
Accept: image/gif, image/x-bitmap, image/jpeg, image/png, */
Accept-Encoding: gzip
Accept-Language: es, en;
q=0.7, fr; q=0.6
Accept-Charset: iso-8859-1, *, utf-8
```

En la petición anterior, el servidor reconoce, por ejemplo, que el cliente prefiere contenidos en español, seguidos de contenidos en inglés y como última instancia, en francés. El parámetro *q* indica la preferencia, y su valor se encuentra entre 0 y 1, mínima y máxima preferencia, respectivamente. También es habitual utilizar el contenido del campo *user agent* para conocer las características correspondientes al navegador utilizado.

Este tipo de negociación basada en el servidor presenta varias desventajas:

- Es imposible determinar por parte del servidor de manera precisa qué contenido será el mejor para cualquier usuario, ya que se desconocen todas las capacidades reales del dispositivo y las que tiene activadas el *user agent*.
- Resulta ineficiente que el cliente esté transmitiendo sus capacidades en cada petición, y mucho más cuando nos encontramos en un entorno móvil donde se dispone de poco ancho de banda.
- Los parámetros deducibles a partir de las cabeceras HTTP, resultan insuficientes para una correcta descripción del dispositivo, p.e. se desconocen la totalidad de preferencias activadas por el usuario en el navegador.

La solución para la creación de contenidos web independientes del dispositivo propuesta por el World Wide Web Consortium (W3C) [2], la recomendación Composite Capabilities/Preferences Profiles (CC/PP) 1.0 [3], es el lenguaje estándar con el que los dispositivos pueden expresar sus capacidades y preferencias de usuario.

Mediante CC/PP es posible la creación de perfiles donde se expresen las características del dispositivo, agente y/o preferencias del usuario. Cuando el cliente realiza una petición al servidor, le indica su perfil CC/PP, el cual es procesado por el servidor y utilizado de guía para la adaptación de los contenidos adecuándose a las características del dispositivo que ha realizado la petición (ver figura 1).

Además, CC/PP ha sido diseñado para ser utilizado en entornos móviles, evitando en cada momento el envío de información redundante. Gracias a CC/PP, como se comprobará a continuación, se obtiene la flexibilidad que no existe en la negociación de contenidos basada en servidor utilizada en HTTP/1.1.



Figura 1. Adaptación de contenidos para distintos tamaños de pantalla

EL PERFIL CC/PP

CC/PP está basado en RDF (Resource Description Framework) [4], otro estándar creado por el W3C. Aunque el objetivo de este documento no es la descripción detallada de RDF, es necesaria una breve introducción a sus principales características para una correcta comprensión de la composición de los perfiles CC/PP.

RDF utiliza XML (Extensible Markup Language) [5] para ser interpretado por los dispositivos y con el fin de definir vocabularios o esquemas. En cada esquema RDF se encuentran las definiciones de los atributos que representarán las propiedades de los dispositivos (en el caso de CC/PP). El modelo RDF [6] se basa en los tres objetos siguientes:

- Recurso. Cualquier elemento que pueda tener URI (Uniform Resource Identifier) [7] es un recurso, incluyendo documentos HTML, páginas web enteras o documentos XML. Todos los elementos descritos mediante expresiones RDF se denominan recursos.

- Propiedad. Una propiedad es un aspecto específico, característica, atributo o relación utilizada para

describir un recurso. Ejemplos de propiedades serían autor o título.

- **Declaración.** Una declaración en RDF está formada por un recurso junto a una propiedad definida y, con su respectivo valor. Estas tres partes son conocidas como sujeto, predicado y objeto, respectivamente.

A través de las declaraciones RDF formadas por tripletas (sujeto, predicado y objeto) es posible la creación de frases de forma similar a cualquier lenguaje. Además, estas frases RDF pueden ser procesadas por los dispositivos para la obtención de su significado. Por esta razón RDF se considera la base para la creación de la web semántica. Un simple ejemplo sería la frase:

" *José Luis Ferrer* es el autor del recurso
<http://www.mat.upc.es/~jferrer/> "

representada en RDF/XML como:

```
<rdf:RDF xmlns:s="http://www.mat.upc.es/~jferrer/
mi_esquema">
  <rdf:Description about="http://www.mat.upc.es/~ferrer/">
    <s:Author>José Luis Ferrer</s:Author>
  </rdf:Description>
</rdf:RDF>
```

El esquema RDF se indica mediante la extensión de XML con la declaración de un XML *namespace*:

```
<rdf:RDF xmlns:s="http://www.mat.upc.es/~jferrer/
mi_esquema">
```

En la URI indicada por el *namespace* XML *s*, http://www.mat.upc.es/~jferrer/mi_esquema, se encuentran las definiciones de los atributos correspondientes al esquema (en el ejemplo se utiliza la propiedad Autor).

Gracias a estar basado en RDF, CC/PP consigue ser flexible y extensible. La flexibilidad la consigue debido a que es posible la definición de un esquema por cualquier usuario, vendedor o fabricante, permitiendo así la descripción de los nuevos dispositivos que vayan apareciendo tan solo añadiendo el nuevo vocabulario. Mientras que la extensibilidad la consigue por otra razón similar, se pueden ir introduciendo nuevos atributos en los vocabularios para describir otras propiedades añadidas a los dispositivos. Además, al estar expresado en XML, distintos dispositivos y elementos (proxies, gateways...) pueden intercambiar descripciones basadas en CC/PP.

El perfil CC/PP, está compuesto por un número de declaraciones en RDF/XML donde se indican los valores de las propiedades utilizadas para definir las capacidades del dispositivo y las preferencias del usuario. CC/PP no proporciona ningún tipo de vocabulario estándar para la composición de los perfiles CC/PP. Sin embargo, gracias a CC/PP se posee el lenguaje para expresar las características de dispositivo y las preferencias del usuario.

El perfil CC/PP se constituye de una jerarquía de dos niveles:

- Un perfil con un número de componentes.
- Uno o más atributos por cada componente existente.

Los componentes equivalen a categorías donde se agrupan distintos atributos, los cuales son considerados como las propiedades. Ejemplos de nombres de componentes serían: *SoftwarePlatform*, *HardwarePlatform* o *Application*. Cada uno de estos componentes tendría sus determinados atributos. Por ejemplo, el componente *Application* podría tener los atributos: versión, nombre de la aplicación, vendedor, tipos de archivo soportados,.... Todos estos atributos estarían declarados en el vocabulario o esquema, referidos mediante los nombres de espacio XML.

Un ejemplo de declaración de varios atributos de un componente en RDF sería:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-
rdf-syntax-ns#" xmlns:ccpp="http://www.w3.org/
2002/11/08-ccpp-schema#" xmlns:ej="http://
www.ejemplo.com/vocabulario#">
  <ccpp:component>
    <rdf:Description rdf:about="http://
    www.ejemplo.com/perfil#Hardware">
      <ccpp:component>
        <rdf:Description rdf:about="http://
        www.ejemplo.com/perfil#Hardware">
          <ccpp:component>
            <rdf:Description rdf:about="http://
            www.ejemplo.com/vocabulario#PlataformaHardware"/>
              <ej:AnchoPantalla>320</ej:AnchoPantalla>
              <ej:AltoPantalla>200</ej:AltoPantalla>
            </rdf:Description>
          </ccpp:component>
        </rdf:Description>
      </ccpp:component>
    </rdf:Description>
  </ccpp:component>
</rdf:RDF>
```

A parte de su expresión en XML, RDF suele representarse en un modelo gráfico, como el mostrado en la figura 2.



ARQUITECTURA

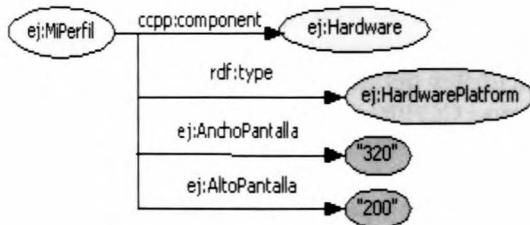


Figura 2. Representación Gráfica RDF

Los atributos de cada componente pueden ser añadidos directamente con su valor en el perfil CC/PP, o pueden ser especificados mediante referencia hacia un perfil por defecto del usuario. Éstos se indican mediante URIs, reduciendo la información a enviar con el perfil, muy importante en entornos móviles. Los atributos de un componente se indican hacia sus valores por defecto mediante *ccpp:defaults* o *ccpp:Default*, definidos en el namespace *ccpp*. En el ejemplo anterior, indicando la URI con los valores por defecto de los atributos de un componente, quedaría como sigue:

```
<ccpp:component>
  <rdf:Description rdf:about="http://www.ejemplo.com/ perfil#Hardware">
    <rdf:type rdf:resource="http://www.ejemplo.com/ schema#PlataformaHardware"/>
    <ccpp:defaults rdf:resource="http://www.ejemplo.com/PerfilHardware#HWDefault"/>
  </rdf:Description>
</ccpp:component>
```

Donde, los valores por defecto de Hardware están en la URI indicada por *ccpp:defaults* expresados mediante RDF:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns" xmlns:ej="http://www.ejemplo.com/ vocabulario#">
  <rdf:Description rdf:about="http://www.ejemplo.com/ PerfilHardware#HWDefault">
    <rdf:type rdf:resource="http://www.ejemplo.com/ vocabulario#PlataformaHardware"/>
    <ej:AnchoPantalla>320</ej:AnchoPantalla>
    <ej:AltoPantalla>200</ej:AltoPantalla>
  </rdf:Description>
</rdf:RDF>
```

Si los atributos de un componente aparecen mediante referencia en su valor por defecto, pero también aparece en el perfil un atributo indicando un valor, el valor añadido dentro del perfil toma preferencia sobre su valor por defecto.

El contexto de uso más simple de CC/PP contemplado por el W3C [8], el mostrado en la figura 3, es el caso en que el cliente envía su perfil de preferencias y capacidades en la cabecera HTTP, aplicando una extensión del protocolo HTTP. El servidor procesa el perfil CC/PP del cliente y le responde con los contenidos adaptados o adecuados según su perfil.



Figura 4. Caso de perfil CC/PP indicado por URI.

Una de las premisas de diseño de CC/PP es su utilización en entornos móviles con dispositivos como PDA's o teléfonos móviles, generalmente hasta el momento en redes de baja capacidad. Es por esta razón, que una de las características más interesantes de las que ofrece CC/PP es la de no enviar siempre el perfil completo, con todas los valores del atributo. Simplemente envía la referencia a un perfil CC/PP indicado por la URI donde se encuentra el perfil. Gracias a este método, se reduce considerablemente la cantidad de información a enviar. Tal y como se puede observar en la figura 4, lo único que debe hacer el servidor es obtener el perfil del repositorio indicado por el cliente. Existe otra ventaja importante de CC/PP, ésta reside en que proporciona un método para indicar en una petición tan solo los atributos que han cambiado desde la última petición.

Los distintos dispositivos que se encuentren entre el cliente y el servidor, como proxies o gateways,

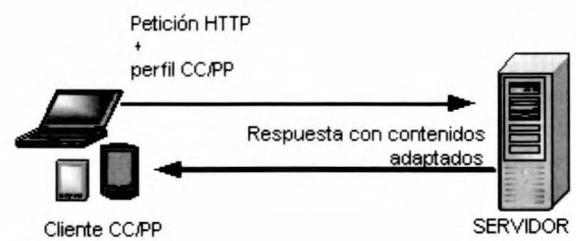


Figura 3. Caso más simple de uso CC/PP

también pueden modificar el perfil CC/PP que había incluido el cliente. Por ejemplo, podrían añadir las preferencia en el tipo de codificación que admiten o cualquier tipo de datos no admitidos por un firewall que existiera entre el cliente y el servidor.

CC/PP EXCHANGE PROTOCOL (CC/PPEx)

CC/PP Exchange Protocol (CC/PPEx) [9] es el protocolo propuesto por el W3C para intercambiar los perfiles CC/PP de la forma más efectiva posible. Este protocolo está basado en el HTTP Extension Framework [10], un mecanismo genérico de extensión para HTTP/1.1, diseñado para ser compatible con las aplicaciones HTTP existentes. A pesar de que éste es un protocolo propuesto por el W3C para la transferencia de descripciones CC/PP, es totalmente independiente de CC/PP y no intenta ser el estándar para el intercambio de información CC/PP.

CC/PPEx se basa en introducir nuevos campos en las cabeceras HTTP. Concretamente se añaden los siguientes:

- Profile. Este campo es una lista de referencias. Cada una representa un objeto correspondiente de la descripción CC/PP. Las referencias de la lista pueden ser URIs absolutas o una codificación MD5 en base64 del valor que se encuentra en un campo *Profile-diff* de la propia cabecera.

- Profile-diff. Empleado para indicar las diferencias en el perfil respecto al perfil anterior. Se permiten varios campos *Profile-diff* en una misma cabecera HTTP de petición. Obviamente, indicar en las cabeceras tan solo las diferencias del perfil respecto la última petición, resulta mucho más eficaz que enviar toda la descripción entera.

- Profile-Warning. Este campo pertenece a la cabecera de respuesta de la extensión de HTTP realizada. Se utiliza para transportar información de aviso, como por ejemplo las indicaciones de transformación de contenidos o no.

A continuación, se muestra un ejemplo de petición:

```
M-GET http://www.ejemplo.com HTTP/1.1
Host: www.w3.org
Man: "http://www.w3.org/1999/06/24-CCPPExchange";
ns=99
99-Profile:"http://www.aaa.com/hw", "http://
www.bbb.com/sw", "1-uKhJE/AEeeMzFSejsYshHg=="
99-Profile-Diff-1: <?xml version="1.0"?>
<RDF xmlns="http://www.w3.org/TR/1999/PR-rdf-
syntax-19990105#" xmlns:PRF="http://www.w3.org/TR/
WD-profile-vocabulary#">
```

```
<Description ID="SoftwarePlatform"
PRF:Sound="On"/>
</RDF>
```

En este caso se trata de una petición obligatoria extremo-a-extremo, ya que se utiliza el método *M-GET* (*Mandatory GET*) y se incluye el campo *Man* en la cabecera. Esta petición indica un diferencial de perfil (*profile-diff*) y dos referencias indirectas. Para referirse al *profile-diff* se ha aplicado el algoritmo MD5 seguido de una codificación en base64 al valor del campo *profile-diff* y, finalmente insertar el número de *profile-diff* al inicio del resultado. Mediante este método cada diferencial de perfil queda totalmente identificado por este valor que se incluye en el campo *Profile*. Examinando tan solo este campo, proxies y gateways pueden realizar la búsqueda en la tabla de caché de manera más eficiente.

UAPROF

User Agent Profile (UAProf) [11] es el estándar desarrollado por Open Mobile Alliance (OMA) [12], antiguo WAP forum. Esta especificación permite a los dispositivos WAP (Wireless Application Protocol) el intercambio de información sobre las preferencias y capacidades (User Agent Profile), entre el cliente WAP, elementos intermedios de la red y el servidor de origen. UAProf se puede entender como el primer desarrollo a gran escala de CC/PP. UAProf utiliza CC/PP para crear un vocabulario estándar que, actualmente, es usado por todos los dispositivos con la especificación 2.0 de WAP [13]. A parte de crear un vocabulario, la especificación de UAProf también incluye protocolos de intercambio de los perfiles y una codificación binaria para éstos, logrando así una transmisión eficiente de la información sobre el enlace inalámbrico.

UAProf se encuentra en un contexto de uso basado en una arquitectura extremo a extremo, tal y como puede comprobarse en la figura 5.

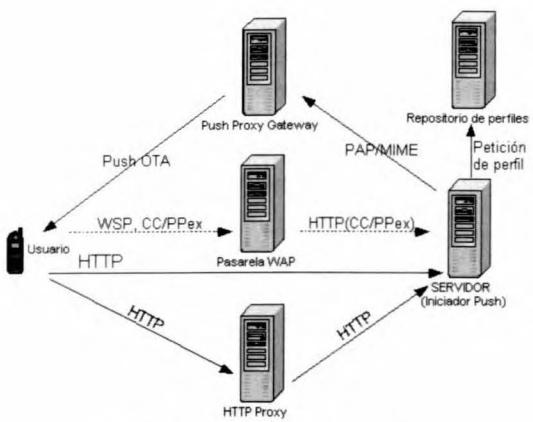


Figura 5. Arquitectura UAProf



Los dispositivos WAP pueden hacer peticiones al servidor de origen mediante la capa de protocolos WAP, o utilizando HTTP directamente sobre el enlace sin hilos, conocido como Wireless Profiled HTTP (W HTTP). Si el cliente se conecta utilizando la capa de protocolos WAP, se hace necesaria la introducción de una pasarela WAP para que realice la petición HTTP al servidor. Opcionalmente, como se observa en la figura 5, puede utilizarse el CC/PP Exchange Protocol (CC/PPex) para la transmisión del perfil.

El servidor es el encargado de indicar al Proxy WAP el envío de los contenidos al cliente sobre el enlace móvil (Over The Air, OTA), utilizando Push Access Protocol (PAP) [14]. La acción Push se utiliza en WAP porque el encargado de entregar los contenidos al cliente no es el servidor, siendo imposible la utilización de un protocolo *stateless*, a base de peticiones y respuestas, como HTTP. Gracias a esta acción, el proxy envía la respuesta al cliente sin que este último haya iniciado la petición al proxy.

Cuando se inicia una sesión WSP (Wireless Session Protocol), el cliente WAP introduce la información de sus preferencias y capacidades en los campos *profile* y *profile diff* de la cabecera de petición de conexión. La pasarela WAP, responde a la petición añadiendo el campo *profile warning* a la cabecera de la respuesta. Si este campo toma el valor 100 (OK), entonces la información del perfil del usuario es cacheada por la pasarela WAP y será válida para todo el tiempo de vida de la sesión. En la especificación de UAPerf, se incluyen los métodos mediante los cuales el cliente puede modificar, suspender o reiniciar su perfil guardado en la sesión. La pasarela WAP, tal y como ocurría en CC/PP, puede añadir información sobre sus características y preferencias al perfil del usuario.

En el caso que la petición del cliente sea sobre W-HTTP, los nuevos campos en la cabecera en los que se transmite la información de las capacidades y preferencias (en caso de que se transmita) son: *x-wap profile* y *x-wap profile diff*. Mientras que el nuevo campo en la cabecera de las respuestas es *x-wap profile warning*.

```
GET http://www.ejemplo.com HTTP/1.1
Host: localhost
x-wap-profile:"http://www.aaa.com/hw","1-uKhJE/
AEeeMzFSejsYshHg=="
x-wap-profile-diff:1; <?xml version="1.0"?>
  <RDF xmlns="http://www.w3.org/TR/1999/PR-rdf-
syntax-19990105#" xmlns:PRF="http://
www.w3.org/TR/WD-profile-vocabulary#">
    <Description ID="SoftwarePlatform"
PRF:Sound="On"/>
</RDF>
```

En cambio, si el transporte de la información CC/PP se realiza sobre WSP, se añaden dos definiciones de nuevos campos en la cabecera de petición y un campo más en la cabecera de respuesta:

- **Profile.** Este campo tan solo contiene una URL (Uniform Resource Locator), que hace referencia a un perfil.
- **Profile-Diff.** Este campo contiene información sobre las preferencias y capacidades codificada en WBXML (Wireless Binary XML).
- **Profile-Warning.** Campo utilizado en las cabeceras de respuesta para indicar incidencias.

Cada petición puede tener varios campos *Profile* y varios *Profile-Diff*. Estas peticiones las realiza el cliente a la pasarela WAP, siendo ésta la que crea las peticiones HTTP, pasando de WSP a HTTP, utilizando la información de la petición y la información de las cabeceras en caché durante la sesión.

La sintaxis especificada para los nuevos campos es la siguiente:

- *Profile* = Número entero(WSP) URI.
El número entero de ocho bits indica el nombre del campo del perfil. Un ejemplo de este campo sería: *Profile*: 0x05 *http://www.aaa.com* 0x00. Donde 0x00 indica el final del string.
- *Profile-Diff* = Número entero(WSP) Longitud Perfil-CCPP.
Por ejemplo: *Profile-Diff*: 0xB6 0x0A 0x01 0x05 0x04..... 0xB6 es el número que indica el nombre del campo de diferencial de perfil, la longitud es de diez octetos (0x0A) y a continuación se incluye la información CC/PP codificada en WBXML.
- *Profile-Warning* = Número entero(WSP) (Código_de_Aviso|longitud Código_de_Aviso URI_con_warning_Fecha)).
En el campo de aviso, si todo ha ido correctamente tan solo se encuentran los valores del número entero identificando al campo y el código de aviso correspondiente a 100:0x90. Mientras que si ha ocurrido algún error en alguna URI, ésta es indicada junto con su código de error y la fecha.

La pasarela WAP, al recibir la petición del cliente, la procesa y transforma a su equivalente HTTP, añadiendo la información CC/PP utilizando el CC/PP Exchange Protocol para indicar las capacidad y preferencias al servidor (ver figura 6). A parte de realizar esta transformación, también realiza caché de cabeceras para determinar los cambios que va realizando el cliente en su perfil. Durante el tiempo de vida de una sesión (WSP), la pasarela WAP realiza la

gestión de las cabeceras para conocer en todo momento el perfil del cliente.

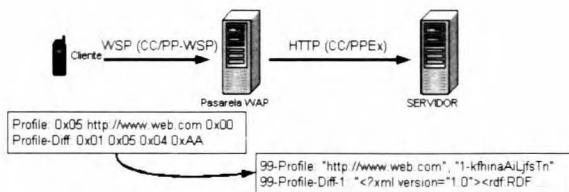


Figura 6. Transformación de CC/PP-WSP a CC/PPEx en la pasarela WAP

Como se ha comentado inicialmente, UAProf proporciona los métodos para que el cliente modifique su perfil durante la sesión, sin necesidad de reiniciarla. La pasarela WAP realiza la caché de cabeceras para crear las peticiones HTTP correspondientes, utilizando información de las cabeceras en caché y de las peticiones que reciba del cliente

A diferencia de la recomendación CC/PP, en UAProf sí se indican los pasos a seguir para la obtención, construcción y modificación de perfiles. En primer lugar, se obtienen los valores de los atributos indicados por sus URIs de referencia. A continuación, se obtienen los valores de los documentos indicados por los campos *Profile* y *Profile diff*. Finalmente, en caso de que aparezcan múltiples descripciones de un atributo, se determina su valor final mediante las reglas de resolución especificadas para cada atributo. Existen tres tipos:

- **Locked**. Indica que el valor que toma el atributo, es el primer valor donde aparece indicado.
- **Override**. La última descripción del atributo es el valor válido.
- **Append**. El valor final consiste en una lista de todas las ocurrencias del atributo en cuestión.

Si no existiera ninguna regla, los valores proporcionados por los diferenciales de perfil (*profile diff*), se sobrescriben sobre los anteriores.

VOCABULARIO UA PROF

El vocabulario especificado por UAProf está compuesto por seis tipos de componentes distintos:

- **HardwarePlatform**. En esta clase se encuentran las propiedades que describen adecuadamente las características hardware del terminal. Entre éstas se incluye, el tipo de dispositivos, tamaño de la pantalla,

- **SoftwarePlatform**. Grupo de propiedades que hacen referencia a la plataforma sobre la que opera el dispositivo. Proporciona información sobre el sistema operativo, los codificadores de video y audio soportados, y las preferencias de lenguaje del usuario.

- **BrowserUA**. Características del navegador web.

- **NetworkCharacteristics**. En este grupo se encuentran los atributos sobre la infraestructura de la red.

- **WapCharacteristics**. Características que pertenecen a las capacidades WAP soportadas por el dispositivo.

- **PushCharacteristics**. Propiedades push específicas que soporta el dispositivo. Por ejemplo, los tipos MIME soportados, el tamaño máximo de mensaje recibido,...

Todos los atributos definidos en el último esquema UAProf se encuentran en:

<http://www.openmobilealliance.org/tech/profiles/cpps schema-20030226.html>

CONCLUSIONES

CC/PP se ha presentado como la solución estándar creada por el W3C para la descripción de las características de cualquier dispositivo. Gracias al hecho de estar basado en RDF, es posible la creación de vocabularios donde se encuentren definidos los atributos (propiedades) agrupados por componentes. En la recomendación no se define ningún vocabulario estándar, sino que se proporciona la plataforma para su desarrollo. Utilizando CC/PP, OMA ha creado UAProf 1.1 como un vocabulario estándar para describir los teléfonos móviles. Aparte de la creación de un vocabulario propio, en UAProf también se especifican unas reglas de resolución a la hora de la composición del perfil CC/PP, ya que la metodología a seguir para la formación del perfil no se define en la recomendación CC/PP.

Ambos estándares, CC/PP y UAProf, exponen los distintos protocolos a utilizar para el transporte e intercambio de perfiles entre los dispositivos y elementos intermedios (como proxies y gateways). La principal premisa de diseño de los protocolos ha sido, obviamente, la optimización de la utilización del canal, ya que los enlaces inalámbricos utilizados para el acceso a Internet no poseen, de momento, un gran ancho de banda y, además, se dispone de baterías limitadas en los dispositivos. El W3C propone el CC/PP Exchange Protocol basado en la extensión de HTTP, como un protocolo que puede ser utilizado (no es el único) para este transporte. Mientras que UAProf especifica el transporte de los perfiles mediante WSP o W-HTTP.



Por otra parte, gracias a RDF, CC/PP consigue ser extensible y flexible, pero esta flexibilidad de CC/PP se convierte en su principal problema. Si se quiere compatibilidad con el mayor número de dispositivos, no pueden existir tantos vocabularios como dispositivos o fabricantes. Por lo tanto es lógico que se piense en definir vocabularios estándares para la descripción de dispositivos del mismo tipo. Asimismo, las definiciones de las propiedades tienen que coincidir si una misma propiedad aparece en varios esquemas. Será necesaria la creación de vocabularios donde una propiedad determinada siempre tenga el mismo significado semántico, de lo contrario será imposible la correcta interpretación de descripciones de distintos dispositivos.

Para finalizar, remarcar la utilidad de CC/PP para su uso en los procesos de adaptación de contenidos, consiguiendo avanzar hacia el acceso universal a la Web, uno de los principales objetivos del W3C. Gracias a su definición, es posible la caracterización de cualquier tipo de dispositivo mediante propiedades definidas en esquemas, que son utilizadas por los servidores para realizar las adaptaciones de contenidos, sin necesidad de almacenar multitud de contenidos para los distintos tipos de dispositivos existentes. UAProf ha sido el primer gran desarrollo de CC/PP, ya implementado en los teléfonos con WAP 2.0. El proceso de elaboración de la recomendación final de CC/PP se ha prolongado durante 5 años, pero la aparición de ésta el pasado 15 de enero de 2004, sirve para que todos las empresas implicadas en la fabricación de dispositivos y desarrollo de aplicaciones independientes del dispositivo puedan implementar, sin controversias, CC/PP.

BIBLIOGRAFÍA Y REFERENCIAS

- [1] Hypertext Transfer Protocol HTTP/1.1. Request for Comments 2616. <http://www.ietf.org/rfc/rfc2616.txt>
- [2] World Wide Web Consortium W3C. [Http://www.w3.org](http://www.w3.org)
- [3] Composite Capabilities/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. W3C Recommendation 15 January 2004 <http://www.w3.org/TR/CCPP-struct-vocab/>
- [4] Resource Description Framework (RDF).
- [5] Extensible Markup Language (XML). [Http://www.w3.org/XML/](http://www.w3.org/XML/) <http://www.w3.org/RDF/>

- [6] Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation 22 February 1999. <http://www.w3.org/TR/REC-rdf-syntax>
- [7] Uniform Resource Identifiers (URI): Generic Syntax. Request for Comments 2396. <http://www.ietf.org/rfc/rfc2396.txt>
- [8] Composite Capabilities/Preference Profiles: Requirements and Arquitechture. [Http://www.w3.org/TR/CCPP-ra](http://www.w3.org/TR/CCPP-ra)
- [9] CC/PP Exchange protocol base on HTTP Extension Framework. [Http://www.w3.org/TR/NOTE-CCPExchange](http://www.w3.org/TR/NOTE-CCPExchange)
- [10] H. Frystyk, P. Leach, S. Lawrence. HTTP Extension Framework. Internet Draft. [Http://www.w3.org/Protocols/HTTP/ietf-http-ext/draft-frystyk-http-extensions-03.txt](http://www.w3.org/Protocols/HTTP/ietf-http-ext/draft-frystyk-http-extensions-03.txt)
- [11] OMA/WAP Forum User Agent Profile 1.1 specification, version 20-October-2001 <http://www.wapforum.org/tech/documents/WAP-248-UAProf-20011020-a.pdf>
- [12] Open Mobile Aliance. <http://www.openmobilealliance.org>
- [13] Wireless Application Protocol (WAP) 2.0 http://www.openmobilealliance.org/tech/affiliates/wap/technical_wap2_0-20020813.zip
- [14] Wireless Application Protocol (WAP) Push Access Protocol Specification, Versión 29 April 2001. WAP-247-PAP-20010429-a

AUTORES



*José Luis Ferrer Riera,
jfer6322@alu-etsitb.upc.es*

Realizando PFC en el Grupo de Redes Inalámbricas del Departamento de Ingeniería Telemática (UPC) sobre la utilización de CC/PP y UAProf para la adaptación de contenidos web en entornos móviles.



*Anna Calveras Augé,
anna.calveras@entel.upc.es*

Dr. Ingeniero de Telecomunicaciones por la Universidad Politécnica de Cataluña (UPC). Profesor Titular de Universidad en el Departamento de Ingeniería Telemática. Forma parte del Grupo de Redes Inalámbricas de dicho departamento.

DESPLIEGUE DE WLANS EN EXTERIORES: DESARROLLO DE UNA HERRAMIENTA PARA LA TOMA DE MEDIDAS GEOREFERENCIADAS

Alexis Porro Pérez^(*), Rafael Vidal Ferré^(**)

(^{*)} Ingeniero Técnico de Telecomunicaciones, especialidad en Telemática. UPC.

(^{**) Departamento de Ingeniería Telemática, Grupo de redes inalámbricas. UPC.}

Contact mail: (^{*)}alexis.porro@estudiant.upc.es, (^{**)rafael.vidal@entel.upc.es}

1. INTRODUCCIÓN

El estándar IEEE 802.11b ha tenido una gran aceptación por parte de los usuarios y ha experimentado un gran *boom* comercial. Esta tecnología estuvo pensada en un principio para dar cobertura a interiores, aunque con el tiempo se ha extendido su uso a exteriores. Un par de ejemplos de este uso los podemos encontrar en la ciudad de Zamora con la empresa Afitel (<http://www.afitel.com>), o en Seattle (<http://seattlewireless.net>) pionera en este tipo de proyectos.

Este hecho provoca la necesidad de obtener medidas de potencia georeferenciadas en exteriores para poder estudiar el perfil de cobertura, y así poder determinar la ubicación de APs o problemas de interferencia entre los APs existentes. Con esta idea en mente se ha diseñado una aplicación gráfica que utiliza una tarjeta WLAN (compatible con el estándar IEEE 802.11b) y un dispositivo receptor de GPS para obtener la medida de potencia georeferenciada en cada punto. Mediante esta herramienta se puede guardar en ficheros toda la información referente a las redes inalámbricas, además de la información de posicionamiento. También ofrece la posibilidad de poder procesar estos datos posteriormente para ser interpretados por la aplicación MatLab.

El artículo empieza explicando como puede obtenerse la información necesaria para realizar medidas efectivas de una red 802.11. En segundo lugar se explica el diseño funcional de la aplicación, separándola en diversos módulos relacionados entre sí. La implementación de estos módulos es explicada a continuación, detallando el software utilizado así como el formato de los ficheros generados. Seguidamente, y a modo de aplicación práctica de la herramienta, se comentan los resultados obtenidos en su utilización para realizar medidas en los exteriores de la EPSC (Escuela Politécnica Superior de Castelldefels) para determinar la cobertura exterior que da la red WLAN instalada en su interior. Para terminar, en las conclusiones se comentan los problemas surgidos durante la implementación de la aplicación así como las líneas futuras.

2. OBTENCIÓN DE INFORMACIÓN DE REDES 802.11

Existen numerosas aplicaciones que permiten obtener información de redes IEEE 802.11, como por ejemplo Kismet, AirSnort, WiFiScanner o PrismStumbler para GNU/

Linux, o NetStumbler para Win32. Todas ellas se basan en gran parte en la escucha de unas tramas de nivel 2 de gestión denominadas beacons, y en la medida de los niveles de la señal recibida mediante una tarjeta 802.11. Estas tramas son emitidas de manera periódica (el periodo de beacon) por parte de los Access Points (APs) a los nodos móviles (STAs) para anunciar su presencia. En esta apartado se comenta el formato de las tramas MAC 802.11 y en concreto de las tramas beacon para conocer toda la información que podemos obtener de ellas.

2.1 Trama MAC 802.11. Los Beacons

La trama MAC queda ilustrada en la figura 1, y se compone básicamente de los siguientes elementos: una MAC Header, el Frame Body de longitud variable y un FCS. El campo de control de trama se compone de los subcampos que aparecen en la figura 2.

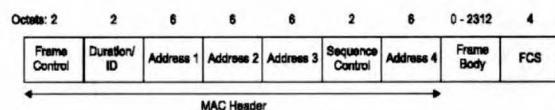


Fig. 1 Formato de la trama

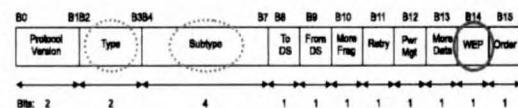


Fig. 2 Campo Frame Control

De estos subcampos, los que tienen especial interés son los que se explican a continuación:

< Subcampos Type y Subtype

El campo de Tipo tiene una longitud de 2 bits, y el de Subtipo de 4 bits. Ambos campos conjuntamente identifican la función de la trama. Existen 3 tipos de trama: las de control, datos y gestión. Dentro de éste último tipo tenemos las tramas de beacon (ver Fig. 3). Cada uno de los tipos de trama tiene diversos subtipos. La tabla 1 define la combinación válida de tipo y subtipo para la trama de beacon.

Tabla 1 Combinación válida de tipo y subtipo para la trama de beacon

Type value b3b2	Type description	Subtype value b7 b6 b5 b4	Subtype description
...
00	Management	1000	Beacon
...



⟨ Subcampo WEP

El campo WEP tiene una longitud de 1 bit y nos permite saber si se está enviando información cifrada utilizando el algoritmo WEP (valor «1») o no (valor «0»).

El formato de una trama de gestión es independiente del subtipo de trama, y está definido en la figura 3.

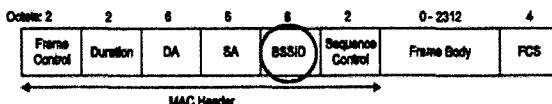


Fig. 3 Formato de la trama de gestión

El Frame Body de la trama de beacon contiene diversos campos fijos que son realmente importantes para la aplicación. Mediante el campo 4, **SSID**, se puede obtener el nombre de la red WLAN o ESSID (Extended Service Set Identifier), y mediante el campo 7, **DS Parameter Set**, se puede obtener el canal utilizado en la comunicación. Otro de estos campos fijos es el *Capability Information*. Este campo 3 tiene una longitud de 2 bytes y queda ilustrado en la figura 4. A partir de los 2 primeros bits de este campo se puede conocer el modo en el que se está trabajando según la tabla 2.

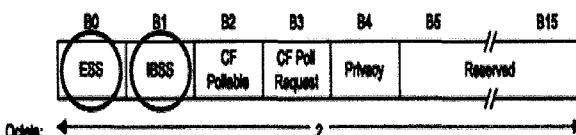


Fig. 4 Campo fijo Capability Information

Tabla 2 Modo de funcionamiento dependiendo de los bits ESS e IBSS

Finalmente, y a modo de resumen, la tabla 3 indica los parámetros que se han obtenido a través de la tarjeta 802.11. Algunos de estos parámetros dependen del contenido de las tramas de beacon, y otros son completamente independientes.

Tabla 3 Parámetros o campos obtenidos a partir de la tarjeta 802.11

CAMPOS

WEP
BSSID
ESSID
CANAL
MODO
POTENCIA SEÑAL
POTENCIA RUIDO
SNR

3. ARQUITECTURA DE LA APLICACIÓN

La herramienta desarrollada ha sido pensada como un conjunto de diversos módulos que interaccionan entre ellos para un correcto funcionamiento de la aplicación. Estos módulos y sus interacciones son descritos a continuación.

3.1 Módulo de WLAN

Este módulo se encarga de la interacción entre la aplicación y la tarjeta WLAN a través de los drivers de ésta. De esta forma, permite obtener los valores de diversos parámetros de la tarjeta WLAN, así como establecer también el valor de algunos de estos parámetros.

Los parámetros que este módulo permite obtener son los que se listan a continuación:

- ⟨ **Canal o Frecuencia**: a la que está trabajando actualmente la tarjeta wireless.
- ⟨ **Bitrate**: velocidad máxima a la que puede transmitir y recibir datos la STA.
- ⟨ **Nickname**: indica el nombre (*nickname*) que tiene la STA para diferenciarla de las demás, aunque es meramente un simple accesorio. De hecho, este parámetro no se usa ya que los protocolos no trabajan con él.
- ⟨ **Signal Level**: potencia de la señal que recibe la STA.
- ⟨ **Noise Level**: potencia del ruido que recibe la STA.
- ⟨ **SNR**: indica la relación señal a ruido de la señal recibida del AP
- ⟨ **ESSID**: identificador de la red ESS en la que está trabajando la STA.
- ⟨ **Dirección MAC del AP**: es el AP al que está asociada la STA en el momento actual.
- ⟨ **WEP**: indica si la transmisión en la red ESS está cifrada (on) o no (off).
- ⟨ **Mode**: modo en el que trabaja la STA o AP, que puede ser: Managed o Ad-Hoc

De ellos, este módulo sólo permite establecer el valor de **Canal o Frecuencia**, **ESSID**, **Dirección MAC del AP** y **Nickname**.

Por otra parte, este módulo tiene dos modos de funcionamiento:

- ⟨ **Normal**: en este modo se puede obtener la información de los parámetros anteriormente citados sobre el funcionamiento actual de la tarjeta. También está implementada una función para poder cambiar los parámetros de la tarjeta.
- ⟨ **De exploración**: en este modo se obtienen los parámetros que caracterizan a un AP. Para esto, se van explorando todos los canales del espectro y se va obteniendo la información que contienen las tramas de gestión de beacon.

3.2 Módulo de GPS

Para este módulo es necesario un dispositivo receptor de

GPS, ya que la aplicación necesita obtener la información de posicionamiento: latitud, longitud y altitud de cada punto. Para tal fin se utiliza una aplicación externa a la propia como interfaz para obtener los datos del dispositivo receptor de GPS.

3.3 Módulo de captura de datos en ficheros

Este módulo permite capturar en ficheros toda la información que obtienen los 2 módulos anteriores (la información referente a la tarjeta WLAN y la referente al dispositivo de GPS), para un posterior estudio de la información obtenida, o un post-procesado de los datos (como se explica en el siguiente módulo). El usuario especifica el nombre del fichero donde se guardan los datos.

3.4 Módulo de tratamiento de capturas

Una vez se tengan los datos en un fichero, se ha pensado en hacer un tratamiento a posteriori de estos. Existen herramientas muy potentes para hacerlo, como el MatLab o sus clones de libre distribución como el Octave o SciLab. Lo que se pretende es aprovechar la potencia de cálculo de estas herramientas transformando los datos a un formato compatible con estas aplicaciones.

Este módulo no interactúa con ninguno de los demás, ya que realiza un post-procesado de los datos capturados por el módulo anterior. Este módulo trabaja con 2 ficheros: el de captura de datos generado por el módulo anterior, y el que será compatible con MatLab, que contendrá la información del fichero anterior de tal forma que la aplicación MatLab sea capaz de interpretarla. Así, a partir de este fichero MatLab, se podrá generar un gráfico en 3D sobre las medidas de potencia georeferenciadas obtenidas.

3.5 Módulo de interfaz gráfica (GUI)

Este módulo concierne toda la interacción con el usuario final a través de los elementos visuales. Está compuesto por elementos típicos de aplicaciones con interfaces gráficas: ventanas, barras de menús, botones, listas deslizables, imágenes en ventanas (como en los mensajes de error) y demás. Con este módulo se consigue la unificación total de la aplicación, ya que a través de la interfaz gráfica se pueden controlar todos los demás módulos.

La figura 5 muestra las relaciones existentes entre los diversos módulos que componen la aplicación. En la siguiente sección se entrará en más detalle en la implementación de esta aplicación.

4. IMPLEMENTACIÓN DE LA APLICACIÓN

Esta aplicación se ha desarrollado completamente bajo una plataforma GNU/Linux. Para la implementación de

esta aplicación sobre un PC ha sido necesario que éste incorpore un adaptador de ISA a PCMCIA (para la comunicación con la tarjeta WLAN) y un puerto serie (para la comunicación con el dispositivo GPS). La potencia del procesador y la cantidad de memoria RAM no son significativas, siempre y cuando el PC sea capaz de ejecutar Red Hat Linux 9.0, la distribución elegida, que incorpora el kernel 2.4.20, con ligereza.

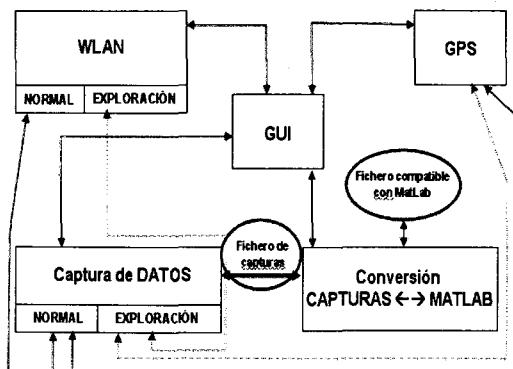


Fig. 5 Diagrama de bloques sobre las relaciones entre módulos

4.1 Módulo de WLAN

Para la realización de este primer módulo se ha utilizado la programación en lenguaje C junto con llamadas al sistema (a través del kernel), haciendo uso de las Wireless Extensions y las Wireless Tools (ambas desarrolladas por Jean Tourrilhes). Se han utilizado algunas de las funciones de las Wireless Tools, otras se han adaptado a las necesidades de la aplicación, y también se han creado nuevas funciones. Básicamente para tal fin se ha utilizado, en primer lugar, un socket (canal de comunicación) entre el kernel y el módulo, mediante el que se ha interactuado con el driver de la tarjeta WLAN. En segundo lugar, y para establecer o conseguir los valores de los parámetros de la WLAN a través del socket anteriormente creado, se ha utilizado la función de C *ioctl (int sock, int request, ...)*. A esta función, y dependiendo si queremos establecer u obtener información de la tarjeta WLAN (get/set), se le pasará una estructura de tipo *wireless_info*, que se define en el fichero *wireless.h*. Este fichero contiene la API de las Wireless Extensions, y normalmente se encuentra bajo el directorio */usr/include/linux/*.

Los drivers PCMCIA de las tarjetas WLAN utilizados en este proyecto son los que se incluyen en la versión 3.2.4 del paquete *pcmcia-cs*. Dependiendo del *driver* y la tarjeta utilizada se han podido obtener y establecer unos u otros parámetros:

- ⟨ Tarjeta Cisco Aironet 350 Series y driver *airo_cs*: permite explorar el espectro en busca de APs mediante los *beacons* que le llegan. Sin embargo no puede obtener la potencia de la señal de ruido, y por tanto

- tampoco la SNR, aunque puede obtener la potencia de la señal.
- ⟨ **Tarjeta Lucent WaveLAN Silver y driver orinoco_cs:** permite establecer todos los parámetros que permite esta aplicación excepto el AP al que asociarse (esta operación no está soportada por su driver). Por otro lado permite obtener la SNR, ya que también puede obtener la potencia de ruido. Sin embargo, el mayor inconveniente que tiene es que no permite explorar el espectro, tan sólo puede obtener la información del AP al que está asociada en un momento determinado.

4.2 Módulo de GPS

El dispositivo receptor de GPS utilizado es un **GARMIN GPS 76**, con diferentes formatos de salida de datos, de los cuales, el que necesitamos es el NMEA 0183, ya que es el único que tiene en común con la aplicación que controla al dispositivo receptor de GPS. Esta aplicación es el demonio de GPS **gpsd**, cuyo autor es Remco Treffkorn (<http://www.pygps.org/gpsd/downloads>). **Gpsd** es un demonio o servidor que obtiene la información de un dispositivo receptor de GPS a través de una interfaz de puerto serie, y espera conexiones de clientes en un puerto determinado (por defecto es el 2947). Esta aplicación espera obtener datos de salida del receptor de GPS en formato NMEA 0183 o en formato binario de Rockwell.

Para interactuar con esta aplicación, se ha hecho una pequeña función que utiliza la comunicación mediante sockets, aunque a diferencia del módulo de WLAN esta vez no se comunica con el kernel, sino con un servidor en un host, que en este caso es la propia máquina que tiene conectado el receptor de GPS.

4.3 Módulo de captura de datos en ficheros

Para este módulo se han utilizado funciones típicas de la gestión de ficheros, como son **open**, **close**, **read** o **write**. En este módulo tan sólo se trabaja con un único fichero en el que se almacenan los datos que se van capturando, y que se obtienen a través de los 2 módulos anteriores.

Para este módulo, se han almacenado los datos pertenecientes a la WLAN junto con los datos que nos ofrece el dispositivo GPS. El formato general del fichero es el siguiente:

- ⟨ 3 bytes de cabecera
- ⟨ datos variables de captura del módulo de WLAN y/o módulo de GPS

Dependiendo de si la tarjeta tiene la capacidad de detectar el ruido o no, hay 2 tipos de formatos de ficheros y datos.

- ⟨ Si la tarjeta puede obtener la potencia de la señal de ruido, entonces podemos conseguir la SNR, y en este caso guardamos la potencia de la señal en dBm, la potencia del ruido en dBm y la SNR en dB. Por tanto, la cabecera se define como **ALL** indicando que se ha guardado todo.

- ⟨ Si por el contrario la tarjeta no puede obtener la potencia de la señal de ruido, no se puede obtener la SNR, por tanto no guardamos ni la potencia del ruido ni la SNR. En este caso la cabecera se define como **SIG** indicando que sólo se ha guardado la potencia de la señal (**SIGnal level**) en dBm.
- ⟨ En el caso que la tarjeta pueda obtener la potencia del ruido (**ALL**), el formato del contenido del fichero es el siguiente:

ESSID SEÑAL	BSSID RUIDO	CANAL SNR	MODO
ENCRYPTACIÓN		LATITUD	
LONGITUD	ALTITUD		

Las siguientes líneas son un ejemplo del formato del fichero de capturas que contiene toda la información posible de la WLAN y además contiene la posición:

```
Essid1      00:11:22:33:44:55      1      -62.00
-90.00 28.00           Managed      on
41.275232  1.987380
```

4.4 Módulo de tratamiento de capturas

En este módulo, al igual que el anterior, se utilizan funciones de gestión de ficheros, así como funciones específicas para cadenas de caracteres o **strings** (incluidas en string.h), como **strstr**, **strrstr**, **index**, **rindex**, etc.

En este módulo, y a diferencia del anterior, se trabaja con 2 ficheros al mismo tiempo:

- ⟨ El de **capturas**, que se abre en modo sólo-lectura
- ⟨ El **compatible con MATLAB**, que se abre en modo sólo-escritura

En este módulo se genera un fichero compatible con MATLAB en el que se crea una matriz con las coordenadas de los puntos (latitud y longitud) para poder obtener una gráfica en 3D, en la que el eje z representa el nivel de señal recibido en dBm o la SNR en dB, dependiendo de las capacidades de la tarjeta utilizada y sus drivers respectivos. En la figura 9 se puede apreciar con más detalle un ejemplo de esta gráfica en 3D.

4.5 Módulo de interfaz gráfica (GUI)

En este módulo, realizado en lenguaje C como todos los demás en esta aplicación, se han utilizado las librerías gráficas de GTK+ en su versión 1.2. Con estas librerías «libres» se ha podido diseñar la parte gráfica de la aplicación, es decir, las ventanas, botones, listas, entradas de texto y otros. Para la implementación de este módulo, se ha utilizado una herramienta de desarrollo gráfico llamada GLADE en su versión 0.6.4. Esta herramienta lo que hace es automatizar y acelerar la construcción de interfaces gráficas utilizando las librerías gráficas GTK+.

En las figuras 6 y 7 se puede observar el aspecto de la

aplicación en pleno funcionamiento tanto en el modo de funcionamiento normal como en el de exploración.

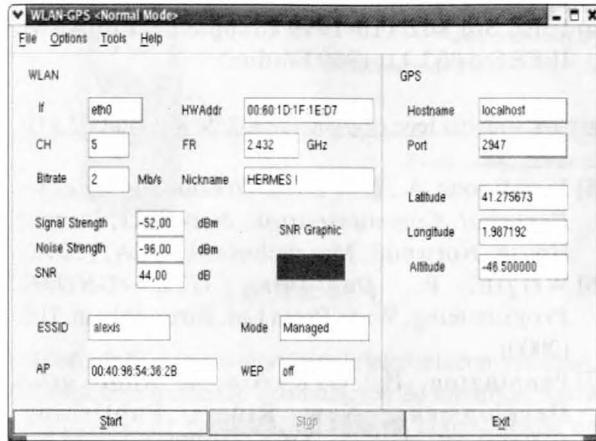


Fig. 6 Ejemplo del modo de funcionamiento normal

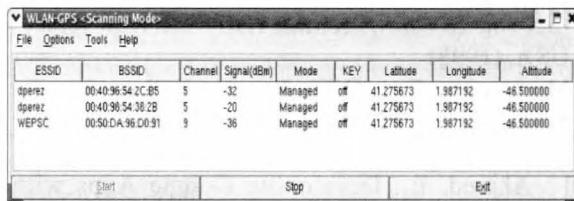


Fig. 7 Ejemplo del modo de funcionamiento de exploración

5. PRUEBAS

Una vez finalizada la implementación de la aplicación, y habiendo comprobado que funciona correctamente, se ha pasado a la realización de capturas de datos en el exterior de la EPSC a modo de ejemplo práctico de las capacidades de la herramienta. Para capturar los datos se ha ejecutado la aplicación en un portátil con la tarjeta WLAN Cisco Aironet 350 Series (con el *driver airo_cs*), que es la que permite explorar el espectro, aunque no permite obtener la potencia de la señal de ruido, y el dispositivo receptor de GPS **GARMINGPS 76**.

Se ha recorrido todo el edificio en 4 pasadas, una por cada pared, haciendo zig-zag para conseguir el mayor alcance posible. El itinerario seguido en las capturas se muestra en la figura 8.

Una vez explorado todo el exterior de la escuela, se ha realizado un post-procesado de los datos mediante la herramienta (para obtener ficheros MatLab), que nos ha permitido conseguir diferentes gráficas en 3D de la cobertura WLAN en el exterior de la Escuela, como la que se muestra en la figura 9. En esta figura se puede observar la cobertura que existe en el exterior de la Escuela para la red ESS «WEPSC», representando sólo la información del canal 1, es decir, filtrando desde la aplicación por ESSID

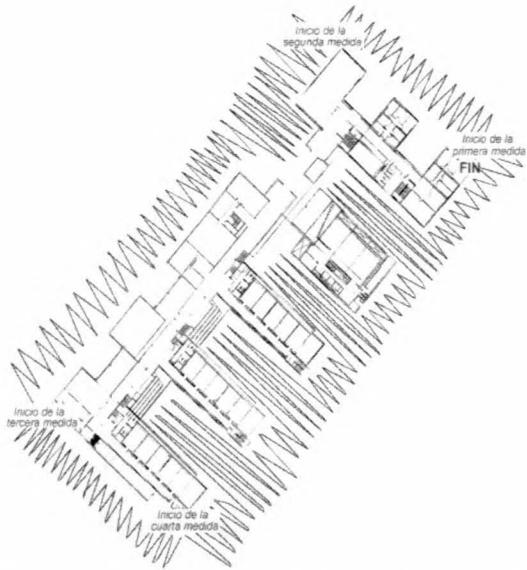


Fig. 8 Mapa que muestra el itinerario seguido para las medidas en el exterior

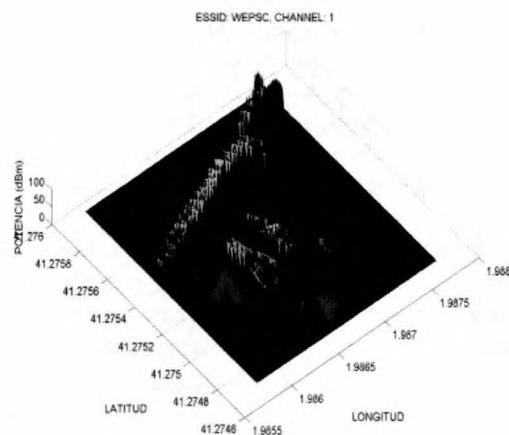


Fig. 9 Gráfica en 3D sobre la cobertura de la red «WEPSC», canal 1

y canal 1. Se puede apreciar el perfil del edificio de la EPSC.

6. CONCLUSIONES

Con el desarrollo de la aplicación se han conseguido diversos objetivos. El primero, la implementación de una herramienta (con página oficial en <http://asterx.upc.es/~alexis/wlan-gps.html>) para la toma de medidas WLAN georeferenciadas, mediante la cual poder guardar la información de redes IEEE 802.11b y la información de posicionamiento en ficheros. El segundo, poder convertir los ficheros de capturas de datos en ficheros MatLab, aprovechando así la potencia y flexibilidad que ofrece la aplicación MatLab para la generación de gráficos en 3D, así como cualquier otra operación matemática con los datos.

Además se han realizado diversas medidas en el exterior de la EPSC a modo de ejemplo de aplicación práctica de la herramienta. Así se puede comprobar que la aplicación es funcional y útil. Para validar la aplicación, se han comparado las medidas obtenidas con esta aplicación con las obtenidas con aplicaciones del fabricante de las tarjetas WLAN, y con los datos que se muestran en el display del receptor de GPS. Sin embargo han habido limitaciones de implementación, ya que no todas las tarjetas que se han utilizado ofrecen las mismas funcionalidades, ni los *drivers* respectivos son lo suficientemente buenos para ofrecer toda la información necesaria.

La herramienta tiene otras aplicaciones prácticas potenciales además de la que se ha querido conseguir en primer lugar. Por ejemplo se podría utilizar para estudiar de manera aproximada el diagrama de radiación de antenas no comerciales y/o experimentales, o también para detectar APs o ESSIDs no registrados en una zona dónde no deberían estar. A pesar que existen diversas aplicaciones con funcionalidades similares a la presente, ésta se diferencia de ellas por las diversas funcionalidades extra que incorpora, como el establecimiento de diversos parámetros de la tarjeta WLAN, y la obtención de ficheros preparados para ser procesados por MatLab.

Como líneas futuras queda la representación de las medidas realizadas en cada punto sobre un mapa real y a escala de cualquier zona. Para esta tarea existen diversas aplicaciones, como por ejemplo la aplicación GPSMap (<http://gpsmap.sourceforge.net/>). También se podría utilizar los valores de altitud en cada punto para realizar una gráfica teniendo en cuenta este parámetro, que es muy importante en zonas donde la orografía no es plana. Además, la aplicación MatLab proporciona un gran abanico de posibilidades para la creación de scripts especializados en, por ejemplo, la búsqueda de puntos con interferencias significativas o con potencias por debajo de un umbral mínimo.

7. AGRADECIMIENTOS

Este trabajo ha sido financiado por el proyecto TIC2003-01748.

8. BIBLIOGRAFÍA

- [1] Porro, A. y Vidal, R., *Despliegue de WLANs en exteriores: Desarrollo de una herramienta para la toma de medidas georeferenciadas*. Trabajo Fin de Carrera , EPSC, Julio 2003.
- [2] Prasad, N. and Prasad A., *WLAN Systems and Wireless IP for Next Generation*

Communications, Artech House, Norwood, Massachusetts, USA, (2002)

- [3] ANSI/IEEE Std 802.11, 1999 Edition
<http://standards.ieee.org/getieee802/download/802.11-1999.pdf>
- [4] IEEE Std 802.11b-1999 (Supplement to ANSI/IEEE Std 802.11, 1999 Edition)
<http://standards.ieee.org/getieee802/download/802.11b-1999.pdf>
- [5] Jamalipour, A., *Low Earth Orbital Satellites for Personal Communication Networks*, Artech House, Norwood, Massachusetts, USA, (1998)
- [6] Wright, P., *Beginning GTK+/GNOME Programming*, Wrox Press Ltd, Birmingham, UK, (2000)
- [7] Pennington, H., *GTK+/Gnome Application Development*, New Riders Publishing, Indianapolis, Indiana, USA, (1999)
- [8] Stevens, W. R., *UNIX Network Programming Volume 1, Networking APIs: Sockets and XTI (Second Edition)*, Prentice Hall PTR, New Jersey, USA, (1998)
- [9] Moritsugu, S. y DTR Bussiness Systems, *La biblia de UNIX*, Anaya Multimedia, Madrid, (1999)
- [10] Ahmed, E., *Developing Gnome Apps with Glade*, (2002)
<http://writelinux.com/glade/index.php>
- [11] Gale, T. and Main, I., *GTK v1.2 Tutorial*, página oficial de GTK, (2000)
<http://www.gtk.org/tutorial1.2/>

AUTORES



Alexis Porro, Ingeniero Técnico de Telecomunicaciones en la especialidad de telemática por la EPSC (UPC) desde el año 2003. Actualmente está cursando el primer curso del segundo ciclo de Ingeniería Superior de Telecomunicaciones en la EPSC y trabaja como becario en el Grupo de redes inalámbricas.



Rafael Vidal, Ingeniero de Telecomunicaciones por la ETSETB (UPC) y profesor del Departamento de Ingeniería Telemática desde el año 2000, con docencia en la EPSC (UPC). Forma parte del grupo de investigación de redes inalámbricas desde el año 1998. Su ámbito de trabajo es el soporte a la movilidad en redes IP. Ha participado en diferentes proyectos de financiación pública y privada. Actualmente trabaja en los proyectos RUBI (Red Ubicua Basada en IP, TIC2003-01748) e I2Cat.

SERVICIOS AVANZADOS DE TELEFONÍA IP MEDIANTE SIP



Antonio Abajo Álvarez

Becario del Departamento de Ingeniería Telemática

Estudiante de Ing. Técnica de Telecommunicación, esp. Telemática. EPSC (UPC)

Sergio Machado Sánchez

Profesor Visitante del Departamento de Ingeniería Telemática

ABSTRACT

La telefonía IP es una evolución de la telefonía convencional, que funciona mediante conmutación de circuitos, hacia conmutación de paquetes, un tipo de conmutación, en principio no pensada para este fin y que gracias al crecimiento de la popularidad de las redes de acceso de banda ancha aparece como una realidad viable. SIP es el protocolo de señalización utilizado en este tipo de telefonía. La incorporación de servidores de aplicaciones SIP permite la implementación de servicios avanzados en el marco de la telefonía IP que incrementan las posibilidades que ofrece la telefonía convencional. Para disponer de un servidor de aplicaciones dentro de un entorno hay que definir los elementos básicos de la arquitectura a la que pertenece tales como la integración con otros servidores de aplicaciones, con servidores SIP, con puertas de enlace, etc. La interacción de todos estos elementos se muestra en la explicación de un ejemplo real.

1 INTRODUCCIÓN

Actualmente, la evolución en el entorno de desarrollo de servicios de Internet se centra en facilitar el proceso de diseño e implementación de las aplicaciones que ofrecen servicios. Ya desde los primeros tiempos de la popularización de Internet hubo múltiples propuestas en cuanto a la aplicación de estrategias que permitiesen alcanzar un sistema capaz de gestionar comunicaciones utilizando los protocolos pertinentes. HTTP (HyperText Transfer Protocol) es el protocolo básico para acceder a la información disponible en la Web. La evolución del entorno de aplicación de dicho protocolo supuso, a medida que las posibilidades iban incrementándose, una complejidad en el desarrollo, como por ejemplo las páginas Web dinámicas, con lo que para realizar un website con una gestión óptima de los contenidos era necesario tener unos conocimientos elevados sobre desarrollo de aplicaciones servidoras (ejecutadas en el servidor).

La simplificación del desarrollo de estas aplicaciones comenzó con la aparición de los CGI (Common Gateway Interface), un estándar para conectar servidores de información, como los servidores Web a aplicaciones externas. Así, un documento HTML que sirve un daemon Web es estático, sin embargo si ante la petición HTTP el

servidor ejecuta cierta aplicación externa la respuesta obtenida por el cliente puede ser dinámica, observándose pues, que esta tecnología añadía "inteligencia" al puro servicio de acceso a documentos alojados en la Web. La idea de los CGI ha ido evolucionando en diversas tecnologías hijas como PHP, ASP, etc., y sobre todo la tecnología Servlet de la plataforma J2EE (Java 2 Enterprise Edition). Los servlets son una interfaz genérica para el desarrollo de aplicaciones servidoras cuya finalidad es la de ofrecer un sistema de servidor dinámico que gestione modularmente el procesado de peticiones HTTP definiendo unas interfaces capaces de independizar la parte de aplicación con la de transporte de la información.

Hoy en día el amplio despliegue de las redes de acceso como ADSL, cable, PLC (Power Line Communication) ofrecen mayor ancho de banda a los usuarios a un precio asequible. Si a eso sumamos la elevada optimización de los algoritmos de compresión y transmisión, se abre la posibilidad de ofrecer servicios de voz y video en tiempo real a través de redes de conmutación de paquetes, redes originalmente no orientadas a dicho fin. En este entorno comienzan a aparecer soluciones de telefonía sobre IP (VoIP, Voz sobre IP). En su proceso de desarrollo de esta tecnología surgen nuevos horizontes a medida que los recursos de la red van incrementándose y que ofrecen algo más que el servicio básico de telefonía. De cara a que un operador pueda ofrecer servicios inteligentes de telefonía sobre IP y aprovechando la experiencia y el éxito de los servlets en el entorno Web, aparece la implementación de esa interfaz genérica para el protocolo SIP (Session Initiation Protocol) utilizado para el transporte de la señalización de las comunicaciones de audio y vídeo.

2 INTRODUCCIÓN A LOS SERVIDORES DE APLICACIONES

Los componentes encargados de la ejecución de las aplicaciones externas que añaden inteligencia a los servicios son los servidores de aplicaciones. Se entiende como servidor de aplicaciones a la entidad capaz de resolver peticiones dinámicamente a través de pequeños programas que se ejecutan en el servidor y que responden en función de los parámetros de dichas peticiones. La Fig. 1 muestra un ejemplo de entrada personalizada a un sitio Web usando HTTP. El servidor recibe la petición HTTP, y ejecuta una



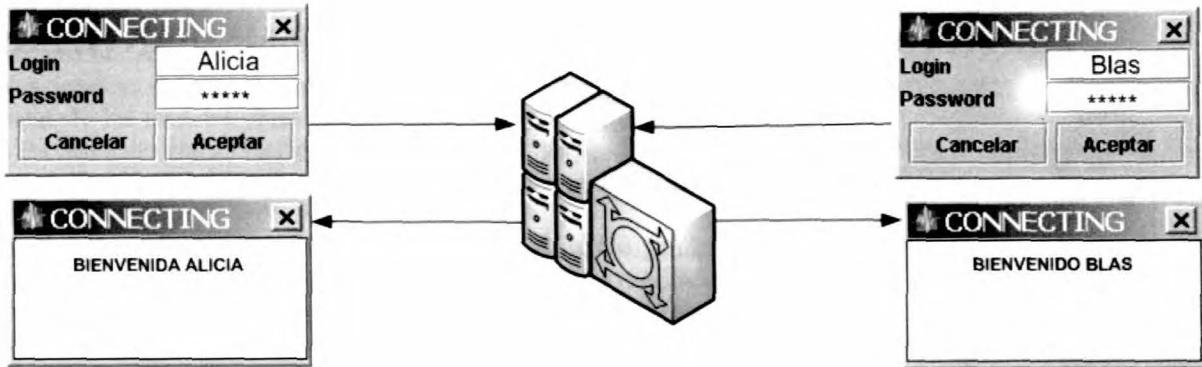


Figura 1 - Ejemplo de funcionamiento de un servidor de aplicaciones

determinada aplicación externa que en función del identificador de usuario y la contraseña entrada por el usuario y transportada dentro de la petición HTTP devuelve una página de bienvenida personalizada que muestra el nombre del usuario que se ha autenticado o, por ejemplo, un error en el caso de que el proceso de autenticación haya fallado por cualquier motivo.

Del mismo modo que sobre HTTP se quería ofrecer respuestas dinámicas en función de los parámetros de la petición, con un servlet SIP se busca ofrecer acciones dinámicas sobre determinados eventos que se pueden producir en el establecimiento o transcurso de una conversación telefónica.

El proceso de estandarización de los servlets SIP se enmarca dentro de una JSR (Java Specification Request). Este proceso consiste en plasmar una idea en un documento como propuesta para que la comunidad JCP (Java Community Process) la evalúe y decida seguir adelante en su estandarización, algo parecido al sistema que sigue el IETF (Internet Engineering Task Force) con los Internet Drafts y los Request For Comments (RFC). La especificación "SIP Servlet API" (JSR número 116) se encuentra en estado "Final" y a partir de aquí, los desarrolladores interesados pueden consultarla para implementarla. En el contenido del documento se encuentra detallado el funcionamiento general del entorno, las clases y métodos que deben estar implementados, y el modo en que se relaciona el contenedor con los "SipServlets" y el entorno. Una JSR proporciona además una implementación de referencia, que tiene por objetivo dar un ejemplo del funcionamiento y composición conceptual del sistema para facilitar la comprensión de la especificación así como crear una implementación que sirva como entorno de pruebas de cumplimiento de la especificación. La implementación de referencia de una JSR no suele ser completamente funcional y no está diseñado para ser usado en entornos de producción.

3 EL PROTOCOLO SIP (SESSION INITIATION PROTOCOL)

SIP es un protocolo creado por el IETF y se encuentra especificado en el RFC 3261. Su propósito es transportar

la señalización de comunicaciones de voz y video a través de una red de conmutación de paquetes. Utiliza unos patrones para definir una lógica de intercambio de mensajes de cara a realizar una serie de funcionalidades que se consideran importantes para posibilitar que dos usuarios en Internet sean capaces de transmitir y recibir flujos de voz y video.

3.1 Situación de SIP en la pila TCP/IP

En la Fig. 2 se puede observar la situación de SIP en la pila TCP/IP. Una implementación SIP debe ser capaz de trabajar a nivel de transporte TCP/IP tanto con TCP (Transmission Control Protocol) como con UDP (User Datagram Protocol), si bien en el RFC se recomienda el uso de UDP por motivos de eficiencia en las comunicaciones y por el hecho de que no es necesario controlar la perdida de mensajes a nivel de transporte ya que SIP posee un control propio.

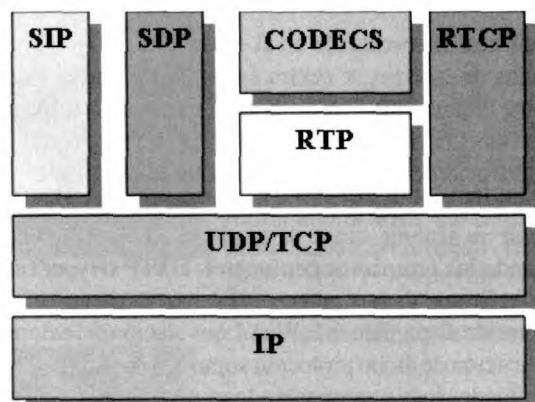


Figura 2 - SIP en la pila TCP/IP

Destacar que el intercambio de información de los flujos de voz y video no viaja sobre el protocolo SIP, si no que para su transmisión se utiliza un protocolo de transporte de información de tiempo real, típicamente Real-time Transport Protocol (RTP). Dentro de la especificación de RTP se incluye la definición del Real-Time Transport Control Protocol (RTCP) para la supervisión de la calidad de servicio durante la transmisión. Otra entidad

que aparece en esta pila son los codecs, necesarios para adaptar los streams de audio y video a un determinado formato. SIP únicamente interviene para señalizar la conversación y para realizar la gestión previa entre los usuarios mediante la cual éstos se intercambian información de configuración de dispositivos, de los codecs que soportan, de la conectividad, etc., necesaria para que la comunicación se realice satisfactoriamente mediante SDP (Session Description Protocol).

Mensajes SIP

#	Tipo	Funcionalidad
1	<i>INVITE</i>	Alicia envía al Servidor SIP un mensaje indicando que quiere llamar a Blas.
2	<i>100 Trying</i>	Servidor SIP responde a Alicia que está intentando establecer la llamada.
3	<i>INVITE</i>	Servidor SIP reenvía el mensaje SIP hacia Blas para indicarle que Alicia le llama.
4	<i>100 Trying</i>	Blas le responde que está procesando la petición.
5	<i>180 Ringing</i>	Blas responde al Servidor SIP que su terminal telefónico está sonando.
6	<i>180 Ringing</i>	El Servidor SIP indica a Alicia que el terminal de Blas suena.
7	<i>200 OK</i>	Blas notifica a Alicia que ha descargado el teléfono.
8	<i>200 OK</i>	Alicia recibe que Blas ha descargado el teléfono.
9	<i>ACK</i>	Mensaje directo de Alicia a Blas previo a la transmisión de voz y video.
10	<i>BYE</i>	Blas cuelga el teléfono y envía este mensaje directo a Alicia.
11	<i>200 OK</i>	Al recibir que Blas cuelga Alicia también cuelga notificándolo con este mensaje.

Tabla 1 - Peticiones SIP

Código de Respuesta	Descripción
1xx	Informativo: respuesta recibida, continuando para procesar la petición.
2xx	Sucesos: Acción recibida correctamente, entendida y aceptada.
3xx	Redirección: Acción futura requerida para completar la petición.
4xx	Error del cliente: Petición contiene una sintaxis errónea o no puede ejecutarse.
5xx	Error del servidor: Error del servidor al ejecutar una petición aparentemente válida.
6xx	Error global: la petición no puede ejecutarse en cualquier servidor.

Tabla 2 - Tipo de respuestas SIP

La Tabla 1 muestra el conjunto de peticiones definidos por SIP, mientras que la Tabla 2 hace lo propio con los mensajes que el usuario puede obtener como respuesta cuando envía una petición. Los mensajes que especifica el estándar SIP no son muy elevados en número en comparación al estándar H.323 que es el que hoy en día está

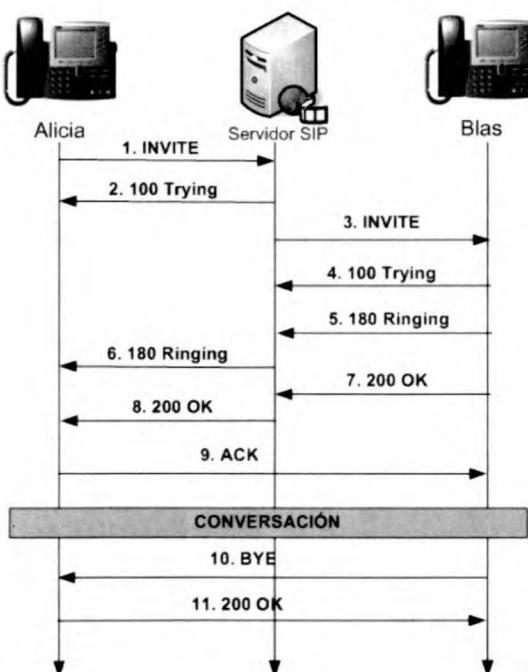


Figura 3 - Establecimiento de una conversación de voz con SIP

vigente en la mayoría de dispositivos destinados a actuar como clientes de voz y video sobre Internet como teléfonos VoIP, aplicaciones de mensajería instantánea tipo Messenger, etc.

La Fig. 3 ejemplariza el inicio de una conversación de voz utilizando SIP y que ayuda a observar el intercambio de mensajes entre los dos comunicantes y el servidor SIP.

La comunicación SIP de la figura 3 se corresponde a un inicio de llamada y a su finalización, ya que la conversación no forma parte del protocolo SIP. En la Tabla 3 se recoge una descripción de los mensajes intercambiados entre los dos comunicantes.

PETICIONES SIP	Descripción
INVITE	El usuario o el servicio está siendo invitado a participar en una sesión.
ACK	El cliente ha confirmado la recepción de un INVITE.
OPTIONS	Para preguntar al servidor sobre sus características o capacidades.
BYE	Indicar que el cliente quiere finalizar la llamada.
CANCEL	Cancela una petición pendiente.
REGISTER	Un cliente registra su dirección en un servidor.

Tabla 3 - Funcionalidad de mensajes del ejemplo

4 ARQUITECTURA DE SERVICIOS AVANZADOS CON SERVLETS SIP

La finalidad de un servlet SIP es posibilitar la programación de servicios avanzados de valor añadido a nivel de aplicación. Como ejemplos de servicios que se pueden desarrollar en este entorno se podrían encontrar lo siguientes: un servicio de presencia (conocer si los usuarios del servicio se encuentran o no disponibles), mensajería instantánea, centralita de telefonía automática, buzón de voz, encaminamiento de llamadas, etc.

Para implementar un servlet SIP, el programador parte de una interfaz ya definida por la especificación. Una interfaz no es más que la declaración de métodos o funciones que ha de realizar una determinada clase o programa por el simple hecho de heredar de ella y que sirve como punto de aislamiento entre clases usuarias de esa interfaz y la implementación de dicha interfaz. Para simplificarlo más una interfaz es la plantilla que han de seguir los programadores para llevar a cabo una parte del programa que estén desarrollando. En la Fig. 4 se recoge el diagrama UML de la interfaz que todo servlet SIP tiene que implementar.

La interfaz de la figura 4 recoge todos los métodos que se pueden implementar en un servlet SIP, entre los más importantes encontramos los métodos *doX()*, donde X es el tipo de mensajes que se recibe en el servidor de aplicaciones. La idea es que toda la programación a partir de este método entrará en funcionamiento en cuanto se reciba un mensaje que origine el evento que redireccione dicho mensaje a ese punto del programa.



SipServlet



Figura 4 - Diagrama UML de la interfaz SIPServlet

4.1 Componentes de la Arquitectura

La Fig. 5 muestra una arquitectura básica de la red de un operador de telefonía IP que usase un servidor de aplicaciones SIP como solución para ofrecer servicios avanzados. Con esta arquitectura un operador puede ofrecer servicios integrados a sus usuarios, servicio de Voz sobre IP e integración con la red de telefonía comunitada

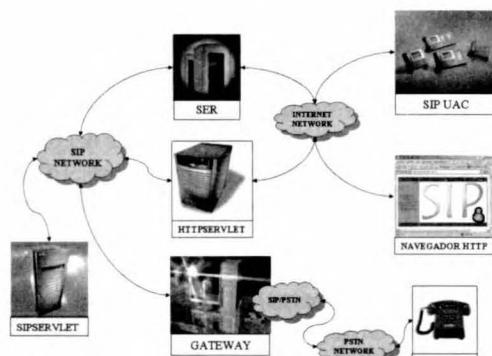


Figura 5 - Arquitectura básica de una red con servicios avanzados SIP

clásica o la red de telefonía de conmutación de paquetes (PSTN, Public Switched Telephone Network).

El operador dispone de un servidor SIP, en este caso se ha utilizado (SIP Express Router). La misión de este dispositivo es la gestión de las llamadas SIP entre agentes de usuario (dispositivos que poseen los usuarios para utilizar los servicios), que funcionen a través de la red de conmutación paquetes mediante la señalización proporcionada por SIP. Para poder comunicar un teléfono IP con abonados de otras redes se necesita disponer de gateways (puerta de enlaces) que adapten la señalización de las dos redes. Para ejecutar los servicios avanzados se dispone del servidor de aplicaciones SIP. Otra entidad que aparece es la de un servidor de aplicaciones HTTP. Tener este componente en la red del operador ofrece la ventaja de su posible integración en la red SIP, ya que las características dinámicas que ofrece HTTP facilitan la convergencia con SIP a través de un adaptador. Con esta arquitectura el operador dispone de una red que integra tanto telefonía convencional como aplicaciones HTTP y SIP.

5 EJEMPLO DE SERVICIO AVANZADO

En el departamento de Ingeniería Telemática estamos desarrollando una aplicación de servicio avanzado de videoconferencia utilizando servlets SIP que denominamos VideoMeeting y en la que existe una convergencia entre SIP y HTTP. Los servicios mínimos que se pretenden proporcionar son:

-**Servicio de Presencia.** Obtener información de la presencia de los usuarios, es decir, si están conectados o no, así como proporcionar información de utilidad como sus datos personales, su dirección de correo electrónico, etc.

-**Creación de reuniones.** Posibilitar la opción de crear reuniones de videoconferencia con los demás usuarios, gestionándose la aceptación de estas reuniones por parte de los usuarios implicados.

-**Vídeoconferencia.** Una vez acordada la videoconferencia para una determinada fecha y hora se establecerá la comunicación entre los usuarios que hayan sido invitados a esa reunión, utilizando el servlet para optimizar la transmisión de los flujos.

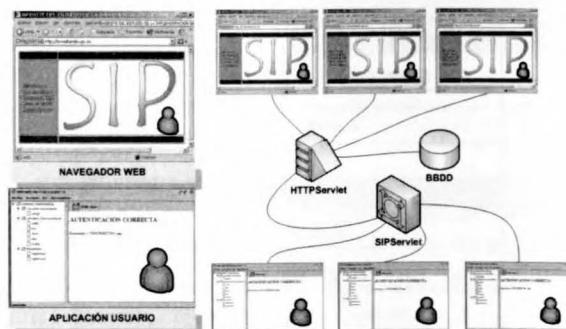


Figura 6 - Esquema de la aplicación Videomeeting

La Fig. 6 representa esquemáticamente el escenario que se persigue en el desarrollo de esta aplicación. Tenemos un servidor de aplicaciones Web con un servlet HTTP que se encarga de ejecutar la lógica de la gestión del servicio de reuniones a través de videoconferencia. Entre otras tareas este servidor se encarga de la autenticación, del registro de usuarios, del alta de reuniones, utilizando como repositorio una base de datos MySQL para almacenar la información de los usuarios. Por otro lado, tenemos un servlet SIP que se encarga de atender las peticiones SIP, funcionando como proxy hacia el servlet HTTP e implementando la lógica de procesado y reenvío de mensajes. Un usuario podrá hacer toda la operativa de gestión tanto a través de cualquier navegador Web como de la aplicación cliente Videomeeting que estamos desarrollando. Esta aplicación, implementada en Java y que utiliza una librería Java de procesado de mensajes SIP llamada NIST (National Institute of Standards and Technology) que permite generar mensajes SIP de forma cómoda, además de, como ya se ha dicho, permitir la gestión, será la que se utilizará para reproducir los flujos de audio, vídeo y texto que envíen el resto de participantes de la reunión.

6 CONCLUSIONES

La creciente popularidad de las redes de acceso de banda ancha proporciona la posibilidad de ofrecer a los usuarios servicios de comunicaciones de voz y video. Gracias a esta capacidad de acceso y a la estandarización de protocolos de señalización como SIP, la telefonía IP se prevé como una alternativa factible a la telefonía convencional. Los servlets SIP, permiten crear servicios avanzados sobre el protocolo SIP lo cual permitirá a los operadores de telefonía IP la gestión y el desarrollo de servicios de valor añadido, dotándose de una herramienta muy potente y sencilla de utilizar, que garantizará dinamismo, eficiencia y un incremento en la calidad de los servicios ofrecidos al usuario.

Para más información:

BIBLIOGRAFÍA:

[1] Practical VoIP using VOCAL, ed. O'Reilly

[1] How TOMCAT works, ed. Brainysoftware

PÁGINAS WEB:

[2] Documentos relacionados con SIP : <http://www.ietf.org/html.charters/sip-charter.html>

- [3] Request For Comments 3261 : <http://www.ietf.org/rfc/rfc3261>
- [4] Java Specification Request 116 (Sip Servlet) : <http://www.jcp.org/en/jsr/detail?id=116>
- [5] Java Specification Request 53 (Java Servlet) : <http://www.jcp.org/en/jsr/detail?id=53>
- [6] JAIN : <http://java.sun.com/products/jain/>
- [7] SLEE tutorial : <http://java.sun.com/products/jain/JAIN-SLEE-Tutorial.pdf>
- [8] SLEE - SipServlet technical comparison : <http://java.sun.com/products/jain/JSLEE-SIPServlet.pdf>
- [9] NIST <http://www.nist.gov/index.html>
- [10] Site Tomcat <http://jakarta.apache.org/tomcat/index.html>
- [11] Site SER: <http://iptel.org>
- [12] Site VOCAL <http://www.vovida.org/vocal>
- [13] Implementación de Referencia Sip-RI : <http://www.sipservlet.org>

AUTORES



Antonio Abajo Álvarez, nacido en L'Hospitalet de Llobregat (Barcelona) en 1980. Obtuvo el título de Técnico Superior en Sistemas de Telecomunicaciones e Informática en 2001. Actualmente está realizando el proyecto final de carrera para la obtención del título de Ingeniero Técnico de Telecomunicación, especialidad Telemática, en la EPSC, mientras trabaja como becario del Departamento de Ingeniería Telemática

Sergio Machado Sánchez, nacido en Barcelona. Obtuvo el título de Ingeniero Superior en Telecomunicaciones en 1998. Trabajó hasta 2003 en IBM España, fecha en la que se incorporó como profesor al Departamento de Ingeniería Telemática impartiendo asignaturas en la EPSC y en la EUETIT. Actualmente está realizando su tesis doctoral centrada en temas de transmisión de vídeo.





REDES AD-HOC: EL PRÓXIMO RETO

Carles Gómez i Montenegro, Josep Paradells Aspas

Wireless Networks Group, Entel Dept., Technical University of Catalonia (UPC)
{carlesgo, teljpa}@entel.upc.es

ABSTRACT

El éxito de las comunicaciones inalámbricas y la progresiva reducción en el tamaño de los dispositivos con capacidad para comunicaciones de datos ha situado en un primer plano a las denominadas redes móviles ad-hoc o MANETs (Mobile Ad-hoc NETworks). En este artículo presentamos las características generales de este tipo de redes, haciendo énfasis en las nuevas condiciones que en ellas se asumen y que constituyen la problemática que la comunidad científica trata de resolver. Presentamos asimismo el estado del arte en temas clave como el encaminamiento y señalamos la familia de aplicaciones de vanguardia que este nuevo paradigma posibilita.

1. INTRODUCCIÓN

¿Qué es una red ad-hoc? Una primera aproximación a la respuesta a esta pregunta radica en el significado de la locución latina que caracteriza a este tipo de redes. El término *ad-hoc* significa literalmente *para esto* [1], con una connotación de *improvisación*. Es decir, una red ad-hoc es un tipo de red que se crea para un cierto propósito de forma temporal. De este modo, en un momento dado, un conjunto de dispositivos independientes puede establecer enlaces

(inalámbricos) entre sí, para cooperar, autoconfigurarse y generar una red, que nace pese a la ausencia de una infraestructura de red previa. La Figura 1 muestra un ejemplo de este concepto.

1.1. Una perspectiva de pasado, presente y futuro

Las redes ad-hoc tienen un origen militar. De forma casi contemporánea al nacimiento de Internet (o de su primer embrión, ARPAnet), a principios de los 70, el ministerio de defensa americano se interesó por el nuevo proyecto Packet Radio Networks (PRNETs) [2]; su objetivo era posibilitar que las distintas unidades en un campo de batalla se pudieran comunicar entre sí mediante dispositivos radio, con libertad de movimiento y de forma cooperativa, de modo que cada nodo podía ser, en cualquier momento, tanto un dispositivo terminal como un conmutador de paquetes.

Durante los primeros años posteriores a su aparición, la investigación en redes ad-hoc ha permanecido en ámbitos militares. Sin embargo, en los últimos años podemos observar dos fenómenos tecnológicos que han tenido un impacto profundo en nuestra sociedad: el éxito de las comunicaciones inalámbricas y la progresiva reducción en el tamaño de los dispositivos con capacidad computacional y conectividad a redes. Este clima ha suscitado el interés masivo de la comunidad académica y científica por las redes ad-hoc, cuyas características suponen un reto para la ingeniería, pero que a su vez posibilitan un gran abanico de nuevas aplicaciones.

2. CARACTERÍSTICAS GENERALES DE LAS REDES AD-HOC

Algunos de los aspectos más relevantes que caracterizan a las redes ad-hoc son los mostrados a continuación.

2.1. Nodos móviles

Los nodos de una red ad-hoc son móviles, pese a que esto no excluye a máquinas fijas, como los ordenadores de sobremesa. De todos modos, se asume que los dispositivos que forman parte de una red ad-hoc pueden cambiar de posición libremente y se comunican entre sí mediante enlaces inalámbricos.

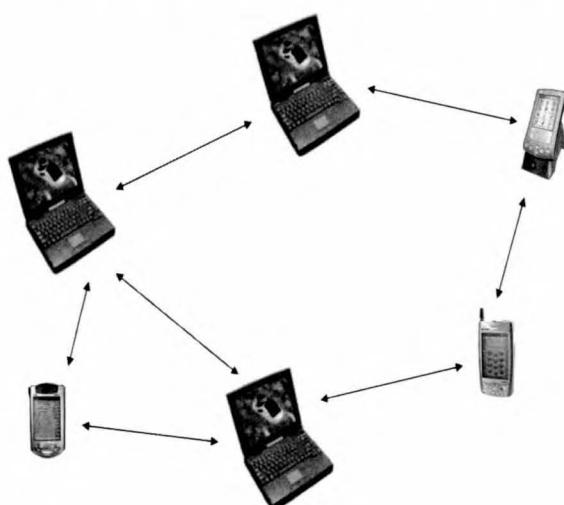


Figura 1: distintos dispositivos con capacidad computacional e interfaz inalámbrica pueden constituir, en un momento dado, una red ad-hoc

2.2. Topología variable

La topología de la red es variable, de forma que un nodo que dispone de un enlace con un nodo vecino puede desplazarse, desaparecer del área de cobertura de su vecino y formar un nuevo enlace con un tercer nodo que caiga dentro de su área de cobertura.

2.3. Cambios de rutas

La rotura de enlaces debida a la movilidad de los nodos provoca que las rutas desde un origen hacia un destino puedan variar con mucha más frecuencia que en redes como Internet. La Figura 2 ilustra con un ejemplo este hecho.

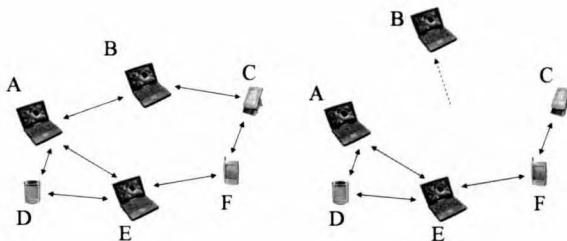


Figura 2: el nodo B se mueve y sale del área de cobertura de los nodos A y C. El tráfico que siguiera la ruta A-B-C deberá utilizar ahora otra ruta, como A-E-F-C

2.4. Dispositivos con capacidad limitada

Un dispositivo móvil debe ser portable y ligero, hecho que implica restricciones a nivel hardware y software. Por otro lado, el tiempo de vida de tales dispositivos viene condicionado por la duración de su batería, que en muchos casos se limita a unas pocas horas [3].

2.5. Limitaciones de los enlaces inalámbricos

La comunicación se efectúa a través de enlaces inalámbricos, que se caracterizan por tener un ancho de banda reducido y ser más propensos a errores que los enlaces fijos [4, 5]. Motivado por la duración limitada de su batería, generalmente transmiten con poca potencia y por ello presentan un limitado alcance que va, dependiendo de las tecnologías, de unos pocos metros a la centena. Este tipo de redes compensan el limitado alcance con la colaboración. Los nodos realizan funciones de repetidor para permitir que dos nodos sin visibilidad directa entre ellos puedan llegar a comunicarse.

2.6. Ausencia de infraestructura

Un cambio significativo frente a los enfoques clásicos en las arquitecturas de redes tradicionales, es que en redes ad-hoc no existe por defecto ningún tipo de entidad centralizada o infraestructura. No se distingue entre dispositivos terminales y enruteadores, de forma que cualquier nodo

puede desempeñar ambos papeles en cualquier momento. Por otro lado, el enfoque clásico de cliente-servidor bajo el cual una gran cantidad de servicios se ejecutan habitualmente en Internet (resolución de nombres, asignación de direcciones, configuración de puerta de enlace, servicios de seguridad, etc) no es válido en una red ad-hoc, porque ningún nodo actúa a priori como servidor.

3. RETOS Y ASPECTOS CRÍTICOS DE LAS REDES AD-HOC

Las características mencionadas en el punto anterior constituyen una problemática con implicaciones en todos los niveles de una comunicación. Por tanto, las redes ad-hoc suponen un nuevo reto, que en algunos casos se podrá afrontar mediante adaptación de los protocolos existentes, mientras que en otros resultará imprescindible crear nuevos mecanismos. En el presente apartado se identifican algunos ámbitos críticos.

3.1. Arquitectura de protocolos

Internet ha aparecido en los últimos tiempos como la Red universal. La tecnología TCP/IP ha demostrado su capacidad de adaptarse a una gran variedad de entornos, aunque en algunos casos haya sido pagando el precio de una cierta degradación en sus prestaciones [4, 6]. Por tanto, TCP/IP es una opción razonable como arquitectura de protocolos para los dispositivos ad-hoc, hecho que además permitiría la integración de los dispositivos ad-hoc en Internet. Sin embargo, ciertos escenarios ad-hoc justifican por sí mismos el uso de una arquitectura de protocolos distinta [7], mejor adaptada a un tipo concreto de necesidades.

Por otro lado, al plantear la idoneidad de una cierta arquitectura de protocolos para los nodos ad-hoc, podemos considerar capas funcionalmente independientes, donde una capa proporciona un servicio a la capa inmediatamente superior (filosofía del modelo OSI), o bien una arquitectura con operación coordinada entre capas cualesquier. Varios estudios demuestran que se pueden obtener beneficios significativos utilizando este último planteamiento en redes ad-hoc [8].

3.2. Nivel de enlace

Existen distintas tecnologías de nivel de enlace que permiten conectividad inalámbrica en modo ad-hoc. En el ámbito de las redes de área local, la familia IEEE 802.11 está experimentando un gran éxito y buena parte de la investigación llevada a cabo en ad-hoc asume el uso de esta tecnología. En el campo de las redes de área personal, de alcance más corto, tecnologías como Bluetooth (o bien IEEE 802.15.1) y la reciente ZigBee (IEEE 802.15.4) parecen estar bien posicionadas para usos distintos. Bluetooth pretende, esencialmente, eliminar los cables entre dispositivos electrónicos domésticos o de uso personal. Por su parte, ZigBee está diseñado como interfaz

inalámbrico para construir redes de sensores con batería de muy larga duración (medida en meses o incluso en años), bajo coste y utilización baja e infrecuente de la red.

3.3. Encaminamiento

Este es uno de los temas centrales en redes ad-hoc. Por este motivo, emplazamos al lector a ver el punto 4 de este mismo artículo, que dedicamos íntegramente a este campo. Para comprender la complejidad del tema, sólo se debe pensar en que cualquier nodo hace funciones de encamionador y que estos pueden cambiar a causa de su movilidad. Los protocolos de encaminamiento deben ofrecer, a pesar de la movilidad de toda la red, un camino de origen a destino, si este existe.

3.4. Nivel de transporte

El protocolo TCP, ampliamente usado en Internet para proporcionar fiabilidad extremo a extremo, sufre una degradación de sus prestaciones cuando se utiliza sobre redes ad-hoc. Esto se debe a los problemas propios de los enlaces inalámbricos, la rotura de estos enlaces debida a la movilidad de los nodos, y la reducción de ancho de banda disponible debido a tráfico de control en la red. Existen varias propuestas de modificaciones y mejoras para el uso de TCP en redes ad-hoc [9, 10, 11]. Por otro lado, existen propuestas de protocolos de transporte nuevos, específicamente diseñados para las peculiaridades de este tipo de redes [12].

3.5. Descubrimiento de servicios

El usuario de una red pretende ejecutar servicios sobre la misma. Por ello, un dispositivo debe poder descubrir qué servicios están disponibles en una red y qué nodos los proporcionan. Teniendo en cuenta las características de las redes ad-hoc, deben plantearse mecanismos que permitan elegir adecuadamente a los nodos responsables de la provisión de un cierto servicio, y asimismo, los procedimientos necesarios para descubrir la existencia de tales servicios por parte del resto de nodos de la red.

3.6. Consumo de batería

Como hemos mencionado, los dispositivos de una red ad-hoc pueden tener una autonomía limitada, debido a su reducida capacidad de batería (ver Figura 3). Por ello, se debe tener en cuenta la implicación que resulte del uso de cualquier mecanismo en cuanto al consumo de energía. Con el ánimo de reducir el consumo al mínimo, los dispositivos sobre los que se construyen los nodos de red deben ser lo más simples posible y con estrategias para que el interfaz radio y el propio programa de control pueda pasar a modo de bajo consumo en caso de inactividad. Pese a que esta consideración es aplicable a todas las capas, algunos de los aspectos más significativos son los siguientes:

3.6.1. Impacto del nivel de enlace y físico en el consumo de batería

Se debe evitar la realización de retransmisiones innecesarias y de colisiones en el canal de acceso, intentar usar *slots* contiguos y pasar a modo reposo siempre que sea posible.

3.6.2. Impacto del nivel de red en el consumo de batería

Reducir la frecuencia de mensajes de control en la medida de lo posible ahorra energía. Asimismo, el nivel de energía disponible en un dispositivo puede ser una métrica relevante a tener en cuenta para el encaminamiento, como indica la propuesta Power Aware Routing (PAR) [13].

3.7. Seguridad

Servicios de seguridad como autenticación, privacidad y disponibilidad deben poder ser proporcionados también en redes ad-hoc. Cifrado y autenticación requieren el uso de claves criptográficas que son difíciles de suministrar sin un control administrativo definido. En cuanto a disponibilidad, nuevos ataques son posibles, como por ejemplo, el agotamiento de batería. Finalmente, los protocolos de encaminamiento ad-hoc deben securizarse.



Figura 3: distintos tipos de PDAs, teléfonos inteligentes y portátiles; estos dispositivos tienen en común su corta duración de batería

3.8. Calidad de servicio

La problemática asociada a garantizar un cierto perfil de calidad de servicio en las redes tradicionales crece sobremanera en redes ad-hoc, a causa de la movilidad de los dispositivos y su capacidad, a priori reducida. Sin embargo, existen varios esquemas que permiten encontrar rutas óptimas de acuerdo con el tipo de servicio que requiera un cierto flujo de datos [14, 15].

4. ENCAMINAMIENTO

El reto que ha cautivado a un mayor número de investigadores en redes ad-hoc y que ha dado lugar a una mayor cantidad de producción científica es la resolución del problema del encaminamiento. Las limitaciones en ancho de banda y las frecuencias de roturas de enlaces no permiten que los protocolos de encaminamiento existentes en Internet para sistemas autónomos (RIP, OSPF) sean adecuados para estos entornos.

En ad-hoc, no existe una única estrategia de encaminamiento posible o válida, sino que podemos encontrar que cada uno de los distintos enfoques realizados hasta el momento puede resultar especialmente adecuado para un tipo de escenario en concreto. Debemos tener en cuenta que los patrones de movilidad de los nodos, las características de los dispositivos a considerar y el tipo de tráfico que éstos deben intercambiar determinarán unas condiciones específicas que pueden diferir significativamente según los casos. De este modo, la paleta de protocolos de encaminamiento es amplia y rica en características.

Los protocolos de encaminamiento en redes ad-hoc pueden dividirse en las siguientes categorías: unicast, multicast y broadcast. En este artículo nos centraremos en los primeros, es decir, los que resuelven cómo llegar desde un origen hasta un único destino.

Una primera clasificación de los protocolos de encaminamiento unicast consiste en identificar dos grandes grupos de protocolos: los proactivos y los reactivos. Por otro lado, existe un conjunto de protocolos híbridos, que combinan características de ambos.

En la actualidad, dos protocolos proactivos (OLSR y TBRPF) han sido estandarizados como RFC. Por su parte, un protocolo reactivo (AODV) también dispone de RFC, mientras que todo apunta a que un segundo protocolo reactivo (DSR) complete dentro de poco el cuarteto de protocolos estandarizados. De este modo, dos mecanismos de cada tipo gozarán de un estatus que les situará como protocolos preferentes para su uso en Internet.

4.1. Protocolos de encaminamiento proactivo

El encaminamiento proactivo busca mantener las tablas de

los nodos permanentemente actualizadas, de forma que cuando un nodo quiera enviar datos a otro, ya disponga de la información necesaria para alcanzar a su destino. En términos generales, esto supone un intercambio periódico de información entre los nodos, y por tanto, la existencia de un overhead de fondo en la red. Sin embargo, la disponibilidad de rutas en cualquier momento permite que el retardo extremo a extremo asociado a la transmisión de un paquete sea bajo, puesto que cuando un nodo quiere mandar un paquete, ya sabe hacia dónde debe mandarlo.

En cuanto al estado del arte de los protocolos proactivos, los siguientes son los dos protocolos más relevantes en la actualidad, siendo el primero el más popular, puesto que la estandarización del segundo se ha producido pocas semanas antes de la elaboración del presente artículo.

4.1.1. Optimized Link State Routing protocol (OLSR)

OLSR [16] es un protocolo de estado del enlace. Su funcionamiento se basa en la elección de nodos especiales denominados MultiPoint Relays (MPRs) que son los encargados de distribuir el tráfico de control por toda la red, indicando los enlaces que existen entre sus nodos. Como se puede observar en la Figura 4. Este esquema mejora sustancialmente la estrategia de inundación clásica, reduciendo el tráfico de control y, por tanto, consumiendo menos ancho de banda. Un nodo descubre a sus vecinos mediante el envío de mensajes *Hello* y calcula sus rutas minimizando el número de saltos de origen a destino.

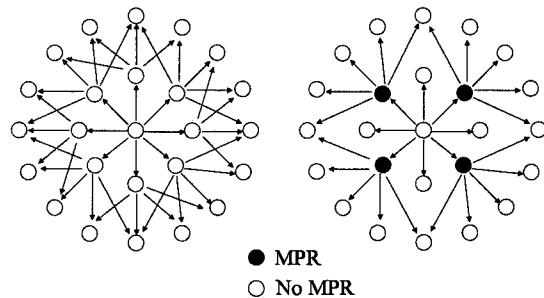


Figura 4: el uso de MPRs permite difundir información de control por toda la red reduciendo el overhead asociado.

4.1.2. Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)

TBRPF [17] es, como en el caso anterior, un protocolo de estado del enlace. Se basa en el cálculo de un árbol con raíz en el nodo origen proporcionando caminos a todos los nodos alcanzables, mediante información parcial de la topología de la red y una modificación del algoritmo de Dijkstra. El protocolo optimiza el descubrimiento de vecinos mediante el envío de mensajes de control diferenciales (envía la diferencia con respecto al anterior), de modo que se minimiza el overhead asociado si no hay cambios en la red.

4.1.3. Otros protocolos proactivos

Como ya hemos mencionado, la colección de protocolos de encaminamiento en redes ad-hoc es grande y sus principios son muy diversos. Algunos ejemplos proactivos que han gozado de relevancia son los siguientes: Destination Sequenced Distance Vector (DSDV), una adaptación para ad-hoc del algoritmo clásico de Bellman-Ford y el protocolo RIP; Source Tree Adaptive Routing (STAR), protocolo de estado del enlace que usa el algoritmo de Dijkstra en cada nodo y Cluster Switch Gateway Routing (CSGR), que agrupa a los nodos en *clusters* e introduce una jerarquía dentro de la red [22, 23].

4.2. Protocolos de encaminamiento reactivo

En oposición al funcionamiento del encaminamiento proactivo, los protocolos reactivos pretenden buscar las rutas bajo demanda entre un nodo origen y un destino. De este modo, cuando un nodo debe mandar un paquete, inicia un mecanismo que le permite averiguar qué camino debe seguir ese paquete. Una desventaja de este planteamiento es el retardo adicional necesario para descubrir una ruta. Sin embargo, el overhead de encaminamiento se puede reducir hasta cero si ningún nodo debe mandar información.

Los siguientes dos protocolos reactivos acumulan una relativa antigüedad, de modo que han sido objeto de simulaciones y pruebas de campo durante los últimos siete años en el primer caso y durante la última década en el segundo.

4.2.1. Ad-hoc On-demand Distance Vector routing (AODV)

AODV [18] consiste en una propuesta planteada para resolver los problemas del protocolo proactivo DSDV. AODV inicia un proceso de descubrimiento de ruta cuando un nodo desea transmitir información. Para ello, como ilustra la Figura 5, difunde paquetes de petición de ruta (RREQ) por la red, hasta que estos alcanzan al propio

destino, o bien a un nodo que conoce cómo llegar al destino. En cualquier caso, este último nodo responde al origen, enviando un mensaje de respuesta (RREP). Los nodos que encaminan la respuesta hacia el origen guardan en sus tablas de encaminamiento la relación entre el nodo originador del RREP y el nodo a través del cual les ha llegado este paquete. Así, cada nodo sabe cuál es su *next hop* para alcanzar un destino dado. Existen ciertos mecanismos de mantenimiento de rutas y la información de las tablas de encaminamiento es borrada tras expirar un tiempo de vida determinado.

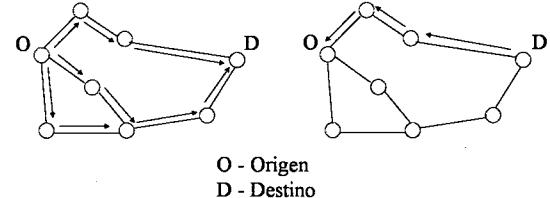


Figura 5: descubrimiento de ruta con AODV; el primer paquete de petición de ruta que llega al destino genera la respuesta por parte de éste

Así, no se mantiene información para nodos inactivos.

4.2.2. Dynamic Source Routing (DSR)

Este protocolo [19] consta de una fase de descubrimiento de ruta bajo demanda y de otra fase de mantenimiento de rutas. La diferencia fundamental con respecto a AODV es que durante el descubrimiento de ruta, los paquetes de petición de ruta almacenan en su cabecera la lista de nodos por los cuales pasan. Cuando uno de estos paquetes alcanza a un destino u otro nodo que sabe cómo llegar al destino, este último nodo envía la respuesta al nodo origen incluyendo igualmente la lista de nodos por los cuales ha pasado la petición de ruta. De este modo, todo paquete de datos transmitido desde el origen, incluye en su cabecera la ruta completa que debe seguir. La Figura 6 presenta un ejemplo de este mecanismo.

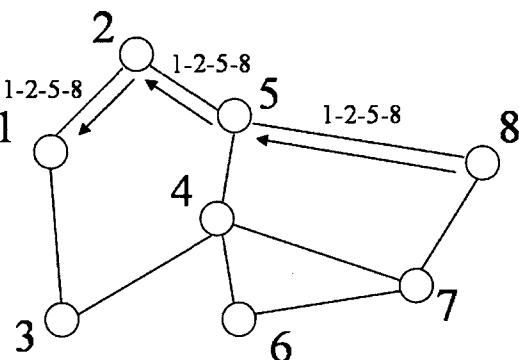
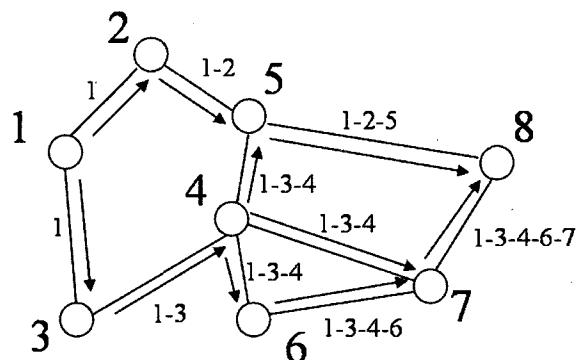


Figura 6: descubrimiento de ruta con DSR; los paquetes de petición de ruta almacenan los nodos por los cuales han pasado

4.2.3. Otros protocolos reactivos

Algunos de los ejemplos más representativos del resto de protocolos reactivos propuestos son los siguientes: Temporally Ordered Routing Algorithm (TORA), un protocolo basado en el concepto de inversión de enlaces; Associativity Based Routing (ABR), basado en el envío de mensajes periódicos breves entre nodos adyacentes para garantizar la asociatividad entre ambos; o por último, Location Aided Routing (LAR), que aprovecha información de localización para reducir la zona en la cual realiza descubrimientos de rutas [22, 23].

4.3. Protocolos de encaminamiento híbrido

Las estrategias reactiva y proactiva son dos puntos extremos en cuanto a concepción de encaminamiento en redes ad-hoc. Sin embargo, existiría un conjunto de grises entre estos dos extremos, que pueden resultar apropiados para ciertos entornos. En esta línea se han definido varias propuestas, la más popular de las cuales es el denominado Zone Routing Protocol (ZRP) [24].

Este protocolo se basa en la definición de zonas en una red ad-hoc, dentro de las cuales se realiza encaminamiento proactivo. Sin embargo, el protocolo funciona de forma reactiva entre zonas.

Una propuesta reciente es el Sharp Hybrid Adaptive Routing Protocol (SHARP) [25], que constituye una mejora frente al ZRP, dado que permite que el radio de las zonas proactivas crezca o se reduzca de forma sensible a la movilidad de los nodos y el patrón de tráfico intercambiado en la red. Esto tiene un impacto significativo con respecto al retardo extremo a extremo y tasa de pérdidas de los paquetes de datos.

5. APLICACIONES Y ESCENARIOS DE USO PARA LAS REDES AD-HOC

Podemos distinguir dos tipos de escenarios para los cuales se prevé el uso de redes ad-hoc: las redes ad-hoc puras, donde no existe infraestructura, y las redes híbridas, que combinan una red ad-hoc con nodos que proporcionan acceso a redes como Internet. Organismos como el IRTF (Internet Research Task Force) han identificado que tales arquitecturas híbridas constituyen los escenarios de uso común con mayor potencial para las redes ad-hoc [20].

5.1. Redes ad-hoc puras

Hay entornos donde la ausencia de infraestructura evidencia la alternativa ad-hoc como única opción para proporcionar comunicaciones entre dispositivos. En otros casos, la existencia de infraestructura es posible, pero supone un coste que con una red ad-hoc resulta nulo. Algunos ejemplos son los siguientes:

5.1.1. Aplicaciones colaborativas

Dispositivos como ordenadores portátiles, PDAs o algunos teléfonos móviles avanzados se han convertido en herramientas de trabajo imprescindibles para varios sectores en el mercado actual. Esta tendencia está alcanzando asimismo a estudiantes y a otros segmentos de la población.

La conectividad ad-hoc permite que los terminales de cualquier grupo de personas puedan construir una red instantáneamente cuando éstas estén concentradas, como por ejemplo en conferencias o reuniones. Así se posibilita el intercambio de información entre los asistentes, o el uso de aplicaciones colaborativas.

5.1.2. Operaciones de rescate y situaciones de emergencia

En algunos casos, establecer comunicaciones mediante los canales habituales como Internet puede no ser posible debido a que tal infraestructura no existe, no es accesible o incluso ha resultado dañada debido a algún accidente o catástrofe. En tal caso, resulta apropiada una solución ad-hoc.

5.1.3. Comunicaciones entre vehículos

Existen proyectos cuya finalidad es dotar con la capacidad de establecer comunicación ad-hoc a vehículos de carretera. Esto permite que tales vehículos puedan intercambiarse información, por ejemplo, para notificar accidentes, cortes de tráfico en la carretera o cualquier tipo de evento imprevisto que deba ser conocido por los conductores.

5.1.4. Redes de sensores

Otra aplicación de las redes sin infraestructura consiste en el uso de una colección de sensores para la realización de medidas. Tales dispositivos pueden crear una red ad-hoc para comunicarse entre sí y transmitir su información. Uno de los proyectos más importantes en esta línea es el Smart Dust de la universidad de Berkeley [21], que ha desarrollado dispositivos que miden estímulos físicos y/o químicos,



Figura 7: distintos tipos de sensores con capacidad de transmisión de datos por interfaz radio: a) sensores MICA2DOT de Crossbow [26] y b) sensor de Smart Dust

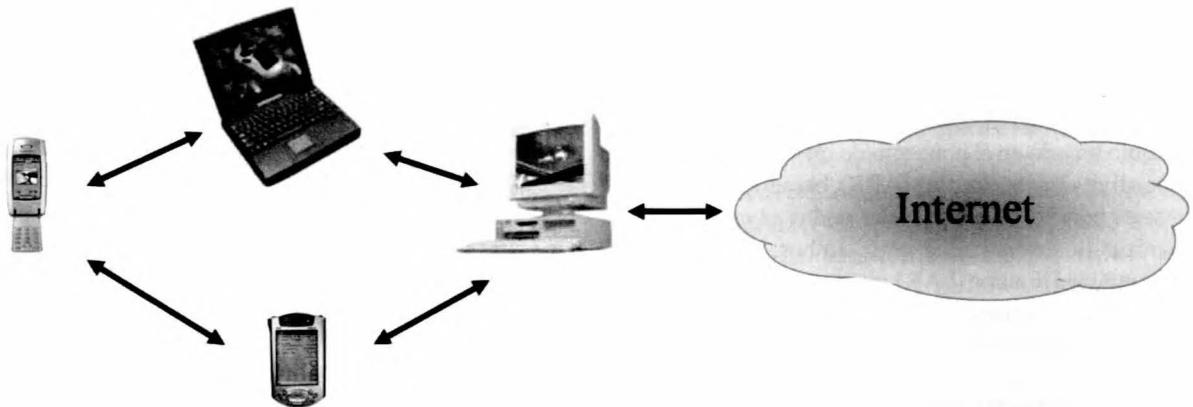


Figura 78: red híbrida; los distintos dispositivos constituyen una red ad-hoc, pero uno de sus nodos (el PC de sobremesa) actúa como nodo de infraestructura proporcionando acceso a Internet

que disponen de capacidad de comunicaciones. El tamaño de tales dispositivos es de aproximadamente un milímetro cúbico. La Figura 7 muestra ejemplos de este tipo de dispositivos.

5.2. Redes híbridas

Las redes híbridas disponen de nodos de infraestructura junto a nodos puramente ad-hoc (ver Figura 8). Dos ejemplos de uso de tales arquitecturas son las redes corporativas y las redes domésticas.

5.2.1. Redes corporativas

Se trata de un tipo de redes destinadas a cubrir uno o más edificios cercanos, en ámbitos como una empresa o un campus universitario. En este escenario existirían los nodos de infraestructura junto a dos tipos de nodos ad-hoc: nodos fijos (alimentados por red eléctrica y con las características de un PC de sobremesa) y nodos móviles (de tamaño reducido como una PDA), con poca batería. Cualquiera de estos dos tipos de dispositivos ad-hoc puede actuar como *router*, pero los primeros disponen de características apropiadas para ser usados con mayor prioridad. Por otro lado, la infraestructura de la red radicaría en puntos de acceso y *gateways* de salida a Internet.

5.2.2. Redes domésticas

Internet está llegando de una forma progresiva a dispositivos de características muy distintas, de modo que la capilaridad de la Red crece. Varias líneas de investigación en domótica apuntan a viviendas cuyos dispositivos están conectados a Internet, de modo que son configurables y accesibles de forma remota. Nos referimos tanto a ordenadores (de sobremesa, portátiles o de bolsillo) como

a electrodomésticos dotados de una cierta inteligencia, e incluso a objetos cuya vinculación con las redes podría parecernos impensable a día de hoy. El uso de interfaces inalámbricas de transmisión y el soporte para conectividad ad-hoc puede resultar una solución clave en estos escenarios.

6. SUMARIO

En este artículo hemos definido el concepto de red ad-hoc y hemos contextualizado este tipo de redes en su marco histórico y socioeconómico. A continuación hemos presentado sus características generales y sus aspectos críticos, indicando varias soluciones planteadas para cada uno de ellos. Hemos tratado con detalle el área del encaminamiento en redes ad-hoc y hemos descrito brevemente los principios de funcionamiento de sus protocolos más relevantes. Finalmente, hemos mostrado los distintos tipos de aplicaciones que se pronostican o empiezan a existir para este tipo de redes, que pueden llegar a convertir la tecnología ad-hoc en un elemento muy presente en nuestras vidas en un futuro próximo.

7. REFERENCIAS

- [1] Diccionario de la Real Academia de la Lengua Española, <http://www.rae.es>
- [2] R. Kahn et al., «Advances in Packet Radio Technology», Proceedings of the IEEE 66:1468-1496, Noviembre 1978.
- [3] RIU253 project, D2.1, «Survey on services, terminals and applications available», <http://www-riu253.upc.es>, 2003.

- [4] RIU253 project, D3.1, «Protocol options and their relation with the wireless packet protocols», <http://www-riu253.upc.es>, 2003.
- [5] A.Calveras, J. Paradells, C. Gómez, M. Catalán, J.C. Vallés, «GPRS and WLAN real networks IP level characterization», IEEE Pacific Rim Conference on Communications, Computers and Signal processing (PACRIM'03), Victoria, Canada, Agosto 2003.
- [6] Hari Balakrishnan, Srinivasan Seshan, Elan Amir, Randy H. Katz. «Improving TCP/IP Performance over Wireless Networks» Proc. 1st ACM Conf. on Mobile Computing and Networking, Berkeley, CA, Noviembre 1995.
- [7] http://wins.rockwellscientific.com/WST_Design_HW.html
- [8] J. Lee, S. Singh, Y. Roh, «Interlayer Interactions and Performance in Wireless Ad Hoc Network», IRTF ANS Working Group Internet Draft, Septiembre 2003.
- [9] J. Liu and S. Singh, «ATCP: TCP for mobile ad hoc networks». IEEE Journalon Selected Areas in Communications, 19(7):1300—1315, Julio 2001.
- [10] G. Holland, N. Vaidya, «Analysis of TCP performance over mobile ad-hoc networks», Proc. ACM/IEEE International Conference on Mobile Computing and Networking, MobiCom'99, Seattle, Washington, USA, Agosto 1999.
- [11] K. Chen, Y. Xue, and Nahrstedt, «On setting TCP's congestion window limit in mobile Ad Hoc networks.» Proc. ICC, Anchorage, USA, 2003.
- [12] K. Sundaresan, V. Anantharaman, H. Hsieh, R. Sivakumar, «ATP: a reliable transport protocol for ad-hoc networks», International Conference on Mobile Computing and Networking, Proc. of the 4th ACM international symposium on Mobile ad hoc networking & computing, Annapolis, Maryland, USA, 2003
- [13] S. Singh, M. Woo, C.S. Raghavendra, «Power-Aware Routing in Mobile Ad Hoc Networks», Proc. of ACM/IEEE MobiCom'98 Conference, October 1998.
- [14] Insignia project, <http://comet.ctr.columbia.edu/insignia/>
- [15] R. Sivakumar, P. Sinha, and V. Bharghavan, «CEDAR: A Core-Extraction Distributed Ad Hoc Routing Algorithm» IEEE Journal on Selected Areas in Communications, Special Issue on Wireless Ad Hoc Networks, vol.17, no.8, pp.1454-1465, Agosto 1999
- [16] T. Clausen, P. Jacquet, «Optimized Link State Routing Protocol(OLSR)», RFC 3626, Octubre 2003
- [17] R. Ogier, F. Templin, M. Lewis, «Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)», RFC3684, Febrero 2004.
- [18] C. Perkins, E. Belding-Royer, S. Das, «Ad hoc On Demand Distance Vector Routing(AODV)», RFC3561, Julio 2003.
- [19] D.B. Johnson, D.A. Maltz, Y.-C. Hu, «The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)», IETF MANET Working Group Internet Draft, Abril 2003.
- [20] L. Yang et al., «Common Wireless Ad Hoc Network Usage Scenarios», IRTF ANS Working Group Internet Draft, Octubre 2003
- [21] Smart Dust project: <http://robotics.eecs.berkeley.edu/~pister/SmartDust>
- [22] C.-K. Toh, «Ad Hoc Mobile Wireless Networks: protocols and systems», Prentice-Hall, 2002
- [23] C. Perkins, «Ad Hoc Networking», Addison-Wesley, 2001.
- [24] Z. Haas, M. Pearlman, P. Samar, «The Zone Routing Protocol (ZRP) for Ad Hoc Networks», Internet Draft, Julio 2002.
- [25] V. Ramasubramanian, Z. Haas, E. Gün Sirer, «SHARP: A Hybrid Adaptive Routing ... In Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (Mobicom)», Annapolis, Maryland, Junio 2003
- [26] Crossbow site, <http://www.xbow.com>

RECONOCIMIENTOS

Este trabajo ha sido financiado en parte por la CICYT TIC2003-01748

BIOGRAFÍA



Carles Gómez i Montenegro es profesor del Departament d'Enginyeria Telemàtica de la UPC en la EPSC (Escola Politècnica Superior de Castelldefels). Ingeniero de telecomunicaciones desde el año 2002 por la ETSETB (Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona), ha participado en proyectos de investigación en el ámbito de las comunicaciones móviles con operadoras como Telefónica y Vodafone, empresas como Flash Networks y el proyecto europeo RIU253 (Recommendations for Internet Usage on 2.5G and 3G). Es coautor de varias publicaciones en congresos nacionales e internacionales y actualmente prepara su proyecto de tesis doctoral en el área de las redes ad-hoc.

TÉCNICAS DE PROCESADO DE SEÑAL BASADAS EN MÚLTIPLES ANTENAS PARA REDES INALÁMBRICAS CON MODULACIONES OFDM



Marcel Palou Agostinho, Antonio Pascual Iserte

Departamento de Teoría de la Señal y Comunicaciones

Universitat Politècnica de Catalunya (UPC)

e-mail: marcel.palou@estudiant.upc.es, tonip@gps.tsc.upc.es

Resumen - En este artículo se discuten las aplicaciones de múltiples antenas en recepción para modulaciones OFDM aplicadas en el ámbito de la red WLAN HiperLAN/2. Se enmarca la problemática y se repasa brevemente las características del sistema. Posteriormente, se discuten diversos algoritmos tanto en el dominio frecuencial como el temporal, para intentar explotar al máximo la limitación de datos finitos de la secuencia de entrenamiento. Se evalúan dichos algoritmos en un simulador de la capa física de HL/2. Se resuelve que las olvidadas técnicas en el dominio temporal mejoran en las prestaciones, imponiéndose a sus competidoras del dominio frecuencial

INTRODUCCIÓN Y ANTECEDENTES

La era de las comunicaciones es ya una realidad en la mayoría de los países, promovida por un rapidísimo aumento de la demanda por parte no sólo de la sociedad en general, sino también del mundo empresarial. Además de las comunicaciones de voz, la principal característica de los sistemas actuales, y de los que aún están por venir en un futuro inmediato, es la interconectividad que se requiere con redes de datos de forma inalámbrica, de manera que se puedan proporcionar servicios de muy diversa naturaleza en cualquier sitio y en cualquier momento. Este último punto muestra uno de los grandes retos en el campo de las radiocomunicaciones, sector donde el espectro radioeléctrico es un bien escaso y preciado. La popular telefonía móvil, con su variante de 3^a generación intenta resolver en parte este problema con un aprovechamiento del espectro mediante técnicas CDMA (Code Division Multiple Access), pese a que su arrancada se prevé complicada y costosa, y las velocidades que se podrán conseguir serán muy moderadas. LMDS (Local Multipoint Distribution System) resuelve con nota el problema de gran ancho de banda, pese a que se queda muy limitado en términos de movilidad. Es en este vacío donde aparecen las WLAN (Wireless Local Area Network), redes cuyo

objetivo inicial era el de proporcionar los mismos servicios que las convencionales LAN, pero sin necesidad de un cableado, y que se están convirtiendo en la alternativa más rentable sobre las otras redes de acceso a datos inalámbricas de alta velocidad. Éstas rompen sin contemplaciones el compromiso entre coste y velocidad, mediante el uso de bandas libres, por otro lado con entornos hostiles en interferencias.

En el sector de las WLAN encontramos el estándar americano IEEE 802.11, una de las redes radio más extendidas actualmente, como resultado de la evolución de la convencional Ethernet (IEEE 802.3). Las velocidades pueden variar bastante hasta conseguir un máximo de 54 Mbps en la variante 802.11a que trabaja a 5.3 GHz. Concretamente, el estándar IEEE 802.11a utiliza la modulación OFDM (Orthogonal Frequency Division Multiplexing), que aprovecha el ancho de banda disponible dividiendo el canal de comunicación en sub-bandas. Este tipo de modulación se combina con una potente codificación de canal (turbo códigos) y el uso de constelaciones de diversa densidad, adecuándose a las condiciones del canal y de la velocidad de transmisión deseada.

A nivel estratégico, Europa necesita el desarrollo también de un estándar de redes WLAN propio, por lo que la ETSI ha desarrollado el HiperLAN/2 (HL/2). A diferencia del estándar americano, HL/2 se enmarca a nivel de arquitectura, entre su homóloga 802.11 y la red de telefonía de 3^a generación. Usa la modulación OFDM, que se caracteriza por ser muy robusta (a interferencias y multi camino) y con una gran eficiencia espectral.

Debido a las dificultades que presenta la transmisión a través del canal radio por la presencia de interferencias, el ancho de banda limitado, los desvanecimientos (también llamados fading), el efecto Doppler, y también a las fuertes y crecientes demandas de tráfico y de número de usuarios en este tipo de redes, es necesario recurrir a sistemas de mejora de la calidad del señal, mediante incremento de la SNR (Relación señal a ruido) y C/I (Relación señal a interferente). Entre las diversas estrategias a seguir, una solución atractiva y que ha llamado la atención de los investigadores en los últimos años es el uso de terminales y/o puntos de acceso con múltiples antenas. El uso de múltiples antenas es muy atractivo a nivel de sistema, ya que gracias a él, se puede permitir un

fuerte rehuso de frecuencias, una reducción del nivel de ruido y sobretodo, una reducción muy elevada del nivel de interferencias (filtraje espacio-temporal). En conclusión, mediante múltiples antenas la cantidad de tráfico que puede cursar la red, y el número de usuarios a los que se les puede prestar servicio de forma simultánea se ve incrementado de forma muy importante.

HIPERLAN/2, DESCRIPCIÓN DEL SISTEMA

HL/2 es un estándar europeo desarrollado por la ETSI para las WLAN [2]. Basado en la modulación OFDM y operando en la banda de los 5.3 GHz (aunque por su modularidad, puede emplazarse en otras bandas con ligeros cambios en la etapa de RF), puede ofrecer velocidades de hasta 54 Mbps en la capa física y de hasta 25 Mbps en la capa de enlace.

El estándar HL/2 especifica una red de acceso radio que puede ser usada con gran variedad de redes núcleo (ATM, Ethernet, ...). Esto es posible gracias a la flexibilidad de la arquitectura, que se basa en independizar las capas físicas y de enlace. Las capas de convergencia específicas para cada red núcleo posibilitan el acceso, y pasan los paquetes de datos de las diferentes redes núcleo a SDU (Service Data Units) con las que trabaja la DLC (Data Link Control) y que asimismo traspasa a las capas de enlace y física. La pila de protocolos se muestra en la figura que sigue.

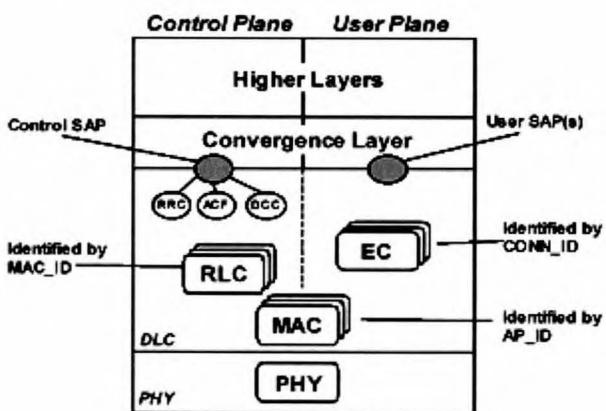


Figura 1: Pila de protocolos de HL/2

El protocolo DLC opera orientado a conexión, creando para cada conexión DLC una instancia EC (Error Control). Las capas de convergencia (CL) son fundamentales para adaptar las peticiones de servicio de las capas superiores al servicio que ofrece la DLC, y convertir los paquetes de capas superiores (pueden ser de longitud fija o variable) en paquetes de longitud fija (SDU) mediante la segmentación y el padding, con los que trabaja la DLC. En la figura que sigue se observa esta convergencia.

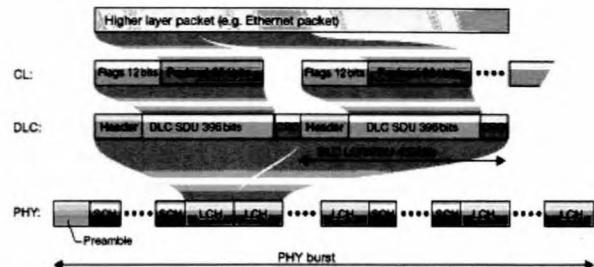


Figura 2: Estructura de tramas del HL/2

Cada SDU está compuesto por 384 bits de datos y 12 bits de flags, que se encapsulan en la DLC en un LCH PDU, con 432 bits (54 bytes) añadiendo una redundancia cíclica (CRC) y una cabecera. Además del LCH (Long Transport Channel), que es el canal de transporte de información, también existe el SCH (Short Transport Channel), que consta de 72 bits por PDU (Packet Data Unit).

La capa de enlace (DLC) está formada por una subcapa de control del enlace radio (RLC, Radio Link Control), un protocolo de control de errores (validación de paquetes por parte del receptor) y un protocolo de control de acceso al medio (control central de recursos, TDMA/TDD). De entre sus funciones destacan todo lo relativo con la asociación/desasociación de la red, análisis y síntesis de los BCCH (Broadcast Control Channel) y FCCH (Frame Control Channel), buffering con las CL, la selección dinámica de frecuencias y el control de potencia.

La capa física (PHY) [3] se encarga de transmitir de forma eficiente las ráfagas de longitud variable provenientes de la DLC, donde éstas constan de un preámbulo y un «payload». El payload se compone de un tren de SCH y LCH PDUs. Como ya se ha comentado anteriormente, en la capa PHY se utiliza la modulación OFDM, debido a su buen comportamiento en medios dispersivos. Las modulaciones de cada subportadora siguen esquemas QAM de distinta densidad. Notar que cada subportadora soporta tasas de bit bajas, pero sumadas se obtienen velocidades de hasta 54 Mbps en capa física. Se definen radiocanales de 20 MHz. El procesado, la modulación y demodulación se realiza de forma digital bajo una frecuencia de muestreo de 20 MHz.

Para relajar las especificaciones de los filtros conformadores, se usan 52 subportadoras de las cuales sólo 48 llevan información (las otras 4 son pilotos). La duración del prefijo cíclico (CP) es de 800 ns, lo que permite buenas prestaciones en canales indoor y outdoor con o sin LOS (línea de vista).

La capa PHY proporciona diferentes modos de trabajo, con diversas tasas de bit, de tipo de codificación de canal (códigos convolucionales de tasa madre $\frac{1}{2}$ adaptables a

tasas de 9/16 y 3/4, en orden de mayor a menor protección) y modulación de subportadoras. Los 7 modos que se ofrecen son los descritos en la figura que sigue.

Mode	Modulation	Code rate	Physical layer bit rate
1	BPSK	1/2	6 Mbit/s
2	BPSK	3/4	9 Mbit/s
3	QPSK	1/2	12 Mbit/s
4	QPSK	3/4	18 Mbit/s
5	16QAM	9/16	27 Mbit/s
6	16QAM	3/4	36 Mbit/s
7	64QAM	3/4	54 Mbit/s

Figura 3: Modo de operación HiperLan/2

Toda ráfaga PHY incluye un preámbulo (distinto según el tipo de canal, control, broadcast, uplink,...), en el caso del uplink, el preámbulo se usa a modo de secuencia de entrenamiento y posibilita la estimación de canal y el ajuste de la frecuencia. Además de estas utilidades, y como veremos en los siguientes apartados, el preámbulo permite diseñar de forma adecuada los parámetros del sistema de múltiples antenas o array.

HL/2 soporta movilidades del terminal como mínimo de 10 m/s. Además incluye mecanismos para poder trabajar en medios dispersivos y con altos niveles de interferencia. De esta forma proporciona una comunicación eficaz con bajos niveles de SNR, manteniendo la QoS, lo que lleva a un compromiso entre el rango y la tasa de bruta de datos [4].

BREVE DESCRIPCIÓN MODULACIÓN OFDM

La multiplexación por división en frecuencia (FDM) es una tecnología para transmitir múltiples señales simultáneamente por el mismo medio. Cada señal va modulada bajo una portadora independiente a una frecuencia determinada. OFDM es una técnica de espectro ensanchado que distribuye el flujo de datos entre un gran número de frecuencias portadoras equiespaciadas, lo que reduce el bit rate de cada subportadora, o lo que es lo mismo, alargando el tiempo de símbolo. Este espaciado concreto proporciona la ‘ortogonalidad’, que permite un solapamiento espectral y que el receptor demodule correctamente cada subportadora por separado. Mediante la IFFT se genera el símbolo temporal a transmitirse. Los beneficios de la OFDM son varios, tiene una alta eficiencia espectral, es muy robusto frente a interferentes, y a la distorsión debido al multitrayecto (que produce ISI). Por la naturaleza de multiportadora, OFDM puede pensarse también como una técnica de acceso múltiple (servicios por portadoras distintas)

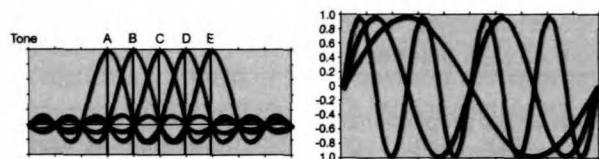


Figura 4: Símbolo OFDM en frecuencia y tiempo respectivamente

Como ya es conocido, la modulación OFDM añade una copia temporal de las últimas muestras al principio de cada símbolo (llamado prefijo cíclico (CP)), que puede interpretarse como una banda de guarda. Evita la ISI y la ICI. En un sistema bien diseñado, su duración debe ser mayor que la longitud efectiva del canal (asegurarnos que el símbolo actual está limpio del anterior). A nivel de cómputo facilita también las cosas, puesto que se puede interpretar una convolución circular entre el canal y el símbolo.

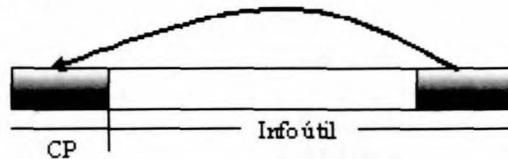


Figura 5: Prefijo cíclico y símbolo OFDM

SISTEMAS CON MÚLTIPLES ANTENAS

Tradicionalmente los sistemas de radiocomunicación se han basado en una antena transmisora y una de receptor, es lo que denomina sistema SISO (Single Input Single Output). Hace ya varios años se desarrollaron sistemas de múltiples antenas para combatir los perniciosos efectos del fading rápido. Un caso típico son los enlaces radio, con 1 antena transmisora y 2 receptoras que se usan de diversidad espacial para combatir los desvanecimientos rápidos. Intuitivamente se denota que si las antenas están suficientemente alejadas, los caminos de la señal deseada para llegar a cada una de ellas serán independientes, por lo que muy difícilmente se verán afectados por fadings simultáneamente.

Técnicas más avanzadas de arrays se basan en la suma o combinación de las señales recibidas por las diversas antenas multiplicando previamente cada una de esas señales por unos coeficientes con un determinado módulo y fase. Mediante esta técnica se pueden generar haces de directividad hacia regiones del espacio predeterminadas, y simultáneamente y de forma contraria, provocar nulos de recepción en ciertas regiones del diagrama de radiación desde las que pueden provenir señales interferentes (Fig. 6).

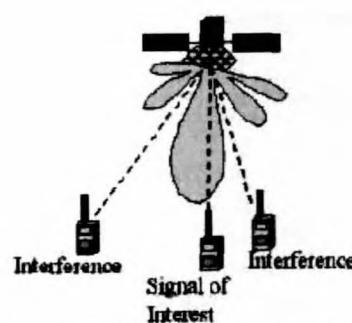


Figura 6 Realce de la señal deseada y cancelación de interferencias

Este diagrama de radiación se controla electrónicamente,

variando su forma mediante el ajuste dinámico de los coeficientes que multiplican a las señales recibidas por las distintas antenas [5]. Tradicionalmente el diseño electrónico del haz se realizaba en la etapa de radiofrecuencia, lo que provocaba difíciles y costosas implementaciones hardware. Actualmente, y gracias al gran rendimiento de los DSP (Digital Signal Processors), esta conformación de haz se realiza en banda base, lo que proporciona una mayor versatilidad, facilidad, y abaratamiento de costes, posibilitando una saga de aplicaciones desconocidas o poco realizables hasta el momento. También debido al procesado en banda base se permite de forma natural conformar diversos beamformers bajo un solo array físico, lo que permite sistemas con múltiples usuarios, con mejora de la señal para cada uno de los usuarios (Fig. 7), o las llamadas Smart Antennas.

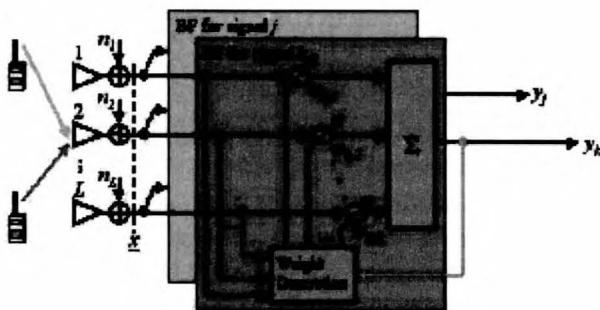


Figura 7 Sistema multi usuario con *smart antennas*

En el entorno de las comunicaciones móviles, donde las interferencias pueden existir, y donde además es deseable que se pueda soportar una gran densidad de usuarios, las técnicas de múltiples antenas han ido vinculadas de forma intrínseca a una mejora de la calidad de la señal, y de forma consecuente, de las prestaciones del sistema. Tradicionalmente se tratan de sistemas SIMO (Single Input Multiple Output) en configuración uplink, con una sola antena en un extremo (terminal móvil) y diversas antenas en el otro extremo (estación base).

El nivel más alto de complejidad corresponde a un diseño conjunto de ambos extremos de la comunicación, con múltiples antenas también en ambos extremos, configurando los denominados sistemas MIMO (Multiple Input Multiple Output). Una posible estrategia de diseño consiste en el cálculo de los coeficientes óptimos a aplicar a la señal antes de transmitirse a través de cada antena transmisora, y de los coeficientes a aplicar a cada una de las señales obtenidas en las diversas antenas receptoras antes de sumarse y llevar a cabo el proceso de demodulación y detección.

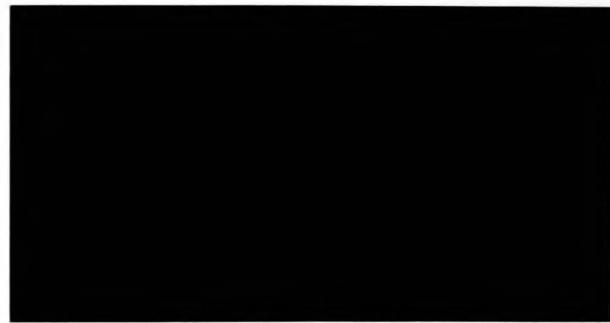


Figura 8: Sistema MIMO

TÉCNICAS SIMO PARA SISTEMAS HL/2 BASADOS EN OFDM – DUALIDAD ESPACIO/TIEMPO

Las redes locales inalámbricas de altas prestaciones se basan en el uso de modulaciones OFDM por su robustez frente a los efectos del multi-trayecto, que provocan ISI (Inter Symbol Interference) e ICI (Inter Carrier Interference), entre otros. Mediante un ajuste adecuado de la longitud del prefijo cíclico o tiempo de guarda, algunos de los efectos anteriores se pueden paliar parcialmente. En la Figura 9 se puede ver la estructura de una trama PHY, y se intuye cómo la duración del prefijo cíclico afecta a la ISI.

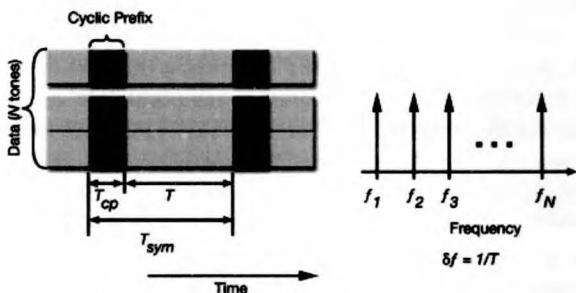


Figura 9: Estructura PHY

En este escrito se pretende describir diversas técnicas SIMO, y mostrar los beneficios obtenidos por éstas aplicadas a entornos de HL/2 basados en modulaciones OFDM. Se mostrarán las técnicas MMSE (Minimum Mean Square Error) descritas en [1] adaptadas a un entorno HL/2 real.

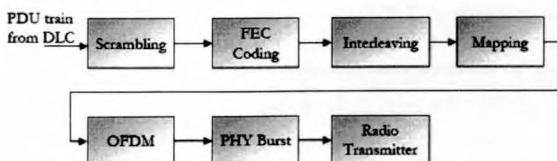


Figura 10: Cadena emisora HL/2 (PHY)

De la dualidad intrínseca frecuencia-tiempo de la modulación OFDM se desprende que el diseño del conformador de haz puede desarrollarse tanto en el dominio temporal, como en el dominio frecuencial. No obstante,

tradicionalmente las técnicas han estado aplicadas en el dominio frecuencial. Dentro de este ámbito, se han evaluado diferentes técnicas basadas en el clásico algoritmo SMI (Sample Matrix Inversion) [5].

Mediante el algoritmo SMI se puede llegar al diseño de un conformador de haz computado mediante la autocorrelación de la señal recibida, y la correlación de dicha señal con una secuencia conocida, típicamente el preámbulo. Mediante esta técnica se diseña el mejor filtro para recuperar la secuencia original a partir de las muestras recibidas. En el caso de secuencias de referencia de longitud infinita, el diseño obtenido coincide la solución óptima de Wiener (ec. 1). En un entorno realista, los datos son finitos, y por lo tanto la matriz de autocorrelación de datos recibidos (R_x) y el vector de correlación cruzada entre datos recibidos y referencia (P) debe estimarse. Debe notarse que el diseño de dicho filtro minimiza el error cuadrático medio aunque no necesariamente maximiza a su vez la SNR.

Wiener $\underline{w} = \underline{R_x}^{-1} \cdot \underline{P}$	SMI $\underline{\hat{R}_x} = \mathbb{E}[\underline{x}_n \cdot \underline{x}_n^T]$ $\underline{P} = \mathbb{E}[\underline{d}(n) \cdot \underline{x}_n]$
	\rightarrow Aprox $\underline{P} = d(n) \cdot \underline{x}_n$

La aplicación de la técnica SMI a un entorno OFDM puede hacerse casi directamente. A nivel lógico, los '0' y '1' de la transmisión se encuentran en el dominio frecuencial, que tras la IFFT (Inverse Fast Fourier Transform) se pasan al dominio temporal que es la información transmitida. Por lo tanto resulta cómodo pensar que este algoritmo pueda aplicarse en el dominio frecuencial.

En un sistema OFDM como HL/2 disponemos de un preámbulo de 2 símbolos OFDM para estimar el canal (también basada en técnicas de mínimo error cuadrático medio, tanto en frecuencia como en tiempo) y llevar a cabo la sincronización, entre otras tareas. El conformador, entendido como el conjunto de coeficientes a aplicar a las señales recibidas para ser sumadas, debe también diseñarse durante la etapa de transmisión del preámbulo (por tener una referencia conocida). Debido a la corta duración del preámbulo, resulta evidente pensar que tendremos un problema de alta variancia en las estimaciones ya que la relación datos vs. parámetros a estimar es baja. Disponemos de 2 símbolos OFDM que corresponden a 128 bits, y el sistema OFDM trabaja con 64 subportadoras de las cuales 52 son útiles, lo que significa que el cálculo del conformador de haz para cada subportadora debe basarse en poco más de 2 muestras, provocando una baja estabilidad de la estimación. Es por ello que se debe pensar en técnicas alternativas para estabilizar las estimaciones. En este trabajo se proponen técnicas de agrupamiento y enventanado de subportadoras.

En el marco temporal de la señal, se presentan soluciones

interesantes y que mejoran las prestaciones del sistema, puesto que en este dominio tenemos más muestras (2 símbolos y el prefijo cíclico), y además debemos estimar menos parámetros (un filtro FIR de M muestras para cada antena). En la figura que sigue se puede ver la cadena receptora para conformadores en tiempo. En este punto se proponen 2 técnicas, el SMI temporal, y el SMI con estabilización semi-ciega añadida.

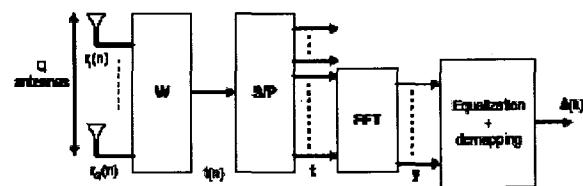


Figura 11: Cadena receptora para técnicas en tiempo

SOLUCIONES EN EL DOMINIO FRECUENCIAL

SMI por agrupación de subportadoras

Consideremos un sistema OFDM SIMO que debe calcular un conformador de haz para cada una de las subportadoras útiles con las que trabaja (en el caso de HL/2 es de 52) puesto que en cada margen frecuencial o portadora el conformador debe responder de forma independiente. No obstante disponemos únicamente de un preámbulo limitado de 128 bits, lo que nos lleva a pensar que no podemos computar de forma fiable los 52 conformadores (alta variancia).

En este subapartado se propone combatir esta baja estabilidad de las estimaciones aprovechándose del ancho de banda de coherencia del canal. En general la respuesta del canal radio es distinta para cada subportadora, y por lo tanto lo óptimo sería diseñar un conformador distinto para cada portadora. Sin embargo, y teniendo en cuenta el concepto de ancho de banda de coherencia del canal [6], se pueden definir grupos de disjuntos de subportadoras, para las cuales el canal variará «poco» o más técnicamente, las respuestas del canal para las subportadoras en un mismo grupo estarán correladas. Para cada grupo de subportadoras se calculará un solo conformador de haz que será usado por todas las subportadoras del grupo.

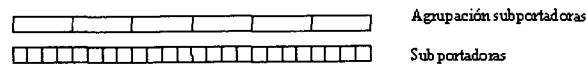


Figura 12: Agrupación de subportadoras

Por lo tanto, las diferentes subportadoras se agrupan en N_g grupos disjuntos de k subportadoras (cada subportadora pertenece a uno y solo un grupo), y para cada grupo se calcula su conformador (beamvector que realiza la señal deseada e intenta anular las interferentes). Existe, de forma innata un grado de libertad, es decir un compromiso sobre el número de agrupaciones a realizar (y

por consiguiente el número k de subportadoras por grupo). Este compromiso nos indica que hay una solución óptima (o mejor compromiso), es decir, que la curva PER (Packet Error Ratio) vs presenta un mínimo local. Recordemos que la PER es un indicativo de las prestaciones del sistema, y está íntimamente ligada al throughput de la capa física. Este compromiso, a alto nivel, puede expresarse de la forma que sigue; si computamos un solo conformador para todas las subportadoras, éste dispondrá de muchos datos para crear unas buenas estimaciones (y P), y generará el conformador que vaya mejor ‘en media’ a todos y lo hará muy bien. Pero debido a que el canal varía subportadora a subportadora, este único conformador se adaptará mal a la variabilidad del canal. En el otro extremo, el cómputo de un conformador para cada subportadora, corregirá muy bien la adaptabilidad del canal, pero por otro lado, estimará muy mal cada conformador.

A nivel técnico, la función de coste a minimizar puede expresarse como la unión de funciones de coste para cada grupo de subportadoras (ec. 2), y por lo tanto tratar cada grupo de forma independiente.

$$\nu^n = \arg \min_{\nu^n} \|X^n \nu^n - \theta^n \mathbf{1}_{k_n}\|^2 \quad (2)$$

donde

ν^n es el conformador para el grupo n.

X^n es la autocorrelación de la parte de muestras recibidas dentro del grupo n.

$\theta^n \mathbf{1}_{k_n}$ es el vector columna con la correlación cruzada dentro del grupo n.

Derivando con respecto a ν^n se obtiene la solución que sigue,

$$\nu_s^n = (X^{nH} X^n)^{-1} X^{nH} \theta^n \mathbf{1}_{k_n} \quad (3)$$

Como podemos observar, formalmente se trata de una solución SMI, adaptada al compromiso antes mencionado, confinando conformadores más suavizados para grupos de subportadoras.

SMI por enventanado de subportadoras

Otra técnica para optimizar el nivel de señal, mediante técnicas de arrays y que se deriva de la intrínseca limitación de datos finitos que conlleva a una complicada estimación de los coeficientes del conformador, es el enventanado de subportadoras.

Tomado el problema enunciado en el apartado anterior, es decir un sistema OFDM, al que intentamos mejorar las prestaciones mediante una arquitectura SIMO, tal y como hemos dicho reiteradamente, la solución óptima es un conformador para cada una de las subportadoras, pero esto no se puede sostener debido a la poca cantidad de datos.

Esta técnica propone crear un conformador para cada subportadora, y aprovechándose también del ancho de banda de coherencia, no solo usar la información de la subportadora actual para realizar la estimación de los parámetros, si no usar datos de sus vecinos mediante un enventanado (diferentes formas de ventana se estudian), de esta forma subportadoras colindantes (frecuencialmente hablando) habrán sido sometidos a canales muy parecidos (recordemos que el canal físico es variante tanto temporalmente, como frecuencialmente, posee un doppler, y puede sufrir desvanecimientos). Se propone pues usar estos datos de subportadoras adyacentes para estabilizar las estimaciones. Normalmente se suelen usar ventanas con un máximo en la subportadora deseada, y caídas hacia los lados, para realzar las subportadoras contiguas a la que se le calcula el conformador, y que las subportadoras lejanas frecuencialmente hablando contribuyan de forma menor.

Es conveniente notar que la información de una subportadora contribuirá a computar distintos conformadores (la información es rehusada).

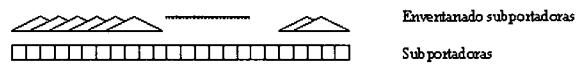


Figura 13: Enventanado de subportadoras

A nivel de la función de coste de dicha solución, deberá parecerse mucho a la encontrada para la técnica precedente. No obstante se generarán conformadores, con un enventanado, o vector de pesos. Se puede observar en este punto que la técnica precedente es un caso particular de esta, donde se hace un muestreo cada k subportadoras y se aplica un enventanado rectangular.

Podemos expresar la función de coste para cada subportadora, por lo tanto la función a minimizar mediante el MSE, como sigue:

$$J(k) = \|\underline{U} \cdot \underline{\chi}^k \cdot \underline{\nu}^k - \underline{U} \cdot \underline{\theta}^k \cdot \underline{1}_{L_u}\|^2 \quad (4)$$

Donde

\underline{U} es la matriz diagonal con los pesos de la ventana.

$\underline{\chi}^k$ es la matriz de muestras del preámbulo recibidas.

$\underline{\nu}^k$ es el conformador a diseñar para la subportadora k.

$\underline{\theta}^k \cdot \underline{1}_{L_u}$ es el vector columna que contiene las muestras conocidas del preámbulo.

Expresando las muestras recibidas enventanadas como $\underline{Z}^k = \underline{U} \cdot \underline{\chi}^k$ y los símbolos transmitidos enventanados como $\underline{\Lambda}^k = \underline{U} \cdot \underline{\theta}^k$, se puede mostrar una solución compacta de la forma que sigue:

$$\underline{\nu}_u^k = (\underline{Z}^{kH} \cdot \underline{Z}^k)^{-1} \cdot \underline{Z}^{kH} \cdot \underline{\Lambda}^k \underline{1}_{L_u} \quad (5)$$



SOLUCIONES EN EL DOMINIO TEMPORAL

Hemos explotado ya alguna solución en el dominio frecuencial para el sistema OFDM. No obstante, pese a su poca utilización actual, el procesado de arrays sobre modulaciones OFDM en el dominio temporal, emerge como una potente base de soluciones que desbanca en prestaciones a sus competidoras del dominio frecuencial. Para explicar este efecto cabe notar que en el dominio temporal disponemos de más muestras (símbolo útil y prefijo cíclico), y también que debemos estimar menos parámetros (filtro FIR de M muestras por antena). El resultado son estimaciones más estables y de mejor calidad, trasladando mejor prestaciones a la globalidad del sistema.

SMI temporal

Primeramente se trata de extender la técnica SMI al dominio temporal. En principio resulta fácil puesto que SMI no entiende de ‘valores’ si no que intenta en todo momento minimizar la distancia cuadrática media entre la distancia recibida y la enviada (MSE) mediante un uso de una secuencia conocida. Por lo tanto el conformador será diseñado bajo una función de coste como la que sigue:

$$\underline{w} = \arg \min_{\underline{w}} \|\underline{\underline{R}}\underline{w} - \underline{\tilde{s}}\|^2 \quad (6)$$

Donde

$\underline{\underline{R}}$ es la matriz de muestras temporales recibidas por las Q antenas

\underline{w} es el vector de pesos del array en el dominio temporal

$\underline{\tilde{s}}$ es el vector de muestras temporales enviadas

Que lleva a una clásica solución SMI como se presenta:

(7)

Hace falta notar que en dicho caso no se necesita ningún tipo de ecualización, y que la complejidad del receptor se reduce dado que solo se necesita un bloque FFT (ver Figura 11). Las muestras se recuperan en el dominio temporal directamente para poderlas introducir en un receptor cualquiera de HL/2.

SMI-CP

Este segundo método intenta explotar al máximo las propiedades de los símbolos OFDM para extraer toda la información conocida. No solo se usa el preámbulo (secuencia conocida) para computar el conformador, si no que se usa de la redundancia cíclica de cada símbolo OFDM para añadir un término ciego adicional de información [7]. Esta técnica, no obstante usa la información de todo el burst (preámbulo + CP de los bits de datos) por lo que trae consigo una latencia de 1 burst.

Se puede deducir pues, que la función de coste que deberá minimizar nuestro diseño del array, tendrá una parte conocida o SMI (en base a un preámbulo conocido), y una parte ciega (en base a la correlación de la información entre el prefijo cíclico y el bloque del cual se ha creado).

$$J_{SBL} = J_{BLI} + J_{SMI} \quad (8)$$

$$J_{BLI} = E \left\| \underline{\underline{R}}^0 \underline{w} - \underline{\underline{R}}^N \underline{w} \right\|^2 \quad (9)$$

$$J_{SMI} = \left\| \underline{\underline{R}} \underline{w} - \underline{\tilde{s}} \right\|^2 \quad (10)$$

Donde aplica a los mismos conceptos que en la técnica precedente. En (ec. 9) puede verse que E se refiere a ‘expectation’, es decir un promedio dentro de todo el burst, de la correlación entre las primeras muestras del símbolo (CP) filtradas, y las últimas también filtradas. Se nota que selecciona las señales que son exactas en los dos bloques temporales (el prefijo cíclico y el bloque que se ha usado para generarla), lo que puede combatir mejor las interferencias, aunque es evidente que la complejidad aumenta sustancialmente.

El conformador óptimo, puede hallarse derivando ec. 8 con respecto al filtro temporal w, y se puede encontrar en [1].

A simple vista parece razonable que este método podrá combatir mejor los efectos de las interferencias, puesto que primeramente obtendrá conformadores más estabilizados (más muestras a tener en cuenta), y puesto que el cálculo del conformador se tiene en cuenta en toda la duración del burst.

CONCLUSIONES

Tras una breve descripción de la arquitectura HL/2, y una introducción a los sistemas de múltiples antenas, se han descrito diversos modos de mejorar las prestaciones de un sistema HL/2 mediante procesado de arrays en recepción, en técnicas que se enmarcan tanto dentro del dominio frecuencial como temporal (aprovechando la dualidad frecuencia-tiempo de OFDM). Se genera pues un filtraje espacio-temporal/frecuencial que permite mejorar sustancialmente las prestaciones del sistema, mediante la cancelación de interferencias, y la densificación de usuarios.

Parece pensar que de forma lógica las soluciones en tiempo son más naturales, y el conformador ataca directamente las muestras sacadas de antena, aunque las soluciones en frecuencia son más entendibles. Es importante notar que las soluciones frecuenciales requieren de una estimación y ecualización del canal, y que requiere una cadena receptora con el bloque FFT para cada antena. En el caso temporal una sola cadena receptora es necesaria, y la ecualización y estimación del canal son implícitas en la técnica. De todas formas el coste computacional resulta más elevado.

En las figuras que siguen se puede apreciar la bondad de

las técnicas propuestas. En la Figura 14 vemos como el procesado de arrays, añadido al MMSE de la estimación de canal, recupera perfectamente la constelación original, tras

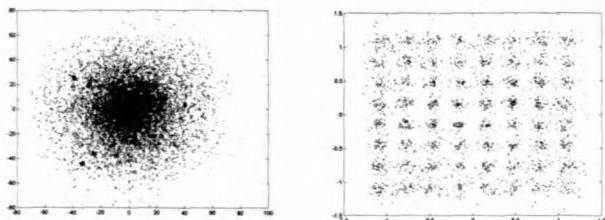


Figura 14: Constelación de las 4 antenas sin procesar, y procesada mediante SMI enventanado

haber recibido una constelación intratable. Los colores se refieren a cada una de las 4 antenas (SNR = 35dB, SIR = 15dB, técnica de frecuencial de enventanado, ventana triangular).

Seguidamente se aprecian las prestaciones. A la izquierda tenemos la PER, en modo 3 con SNR 20. Dada una PER tan baja, la simulación no tiene resolución. A la derecha vemos

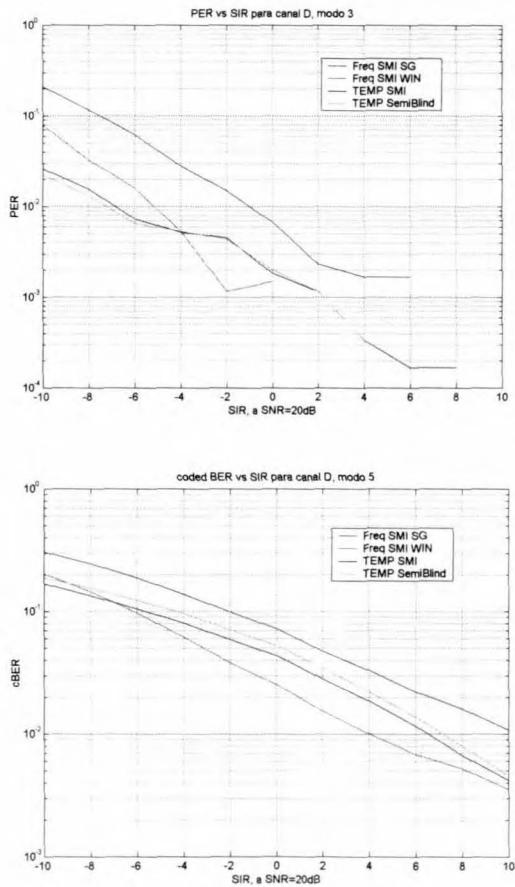


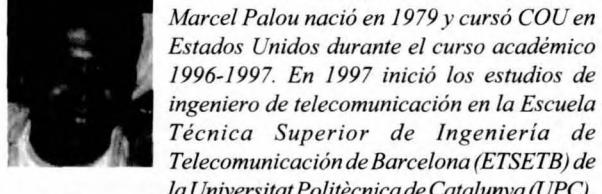
Figura 15: Simulaciones de prestaciones del sistema

la VER después del bloque de codificación de canal para el modo 5 a la misma SNR. Para este caso no se comporta como lo esperado, pero se puede ver que el modo de trabajo para estas condiciones ambientales es degradado.

REFERENCIAS

- [1] D. Bartolomé, A. I. Pérez-Neira, MMSE Techniques for Space Diversity Receivers in OFDM-based Wireless LANs, IEEE Journal of Selected Areas in Communications, February 2003
- [2] ETSI TR 101 683, Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; System Overview, v1.1.1, 2000-02, available at www.etsi.org.
- [3] ETSI TS 101 475, Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Physical (PHY) layer, v1.2.2, 2001-02, available at www.etsi.org.
- [4] Khun-Jush, J. & Malmgren, G. & Schramm, P. & Torsner, J., HIPERLAN type 2 for broadband wireless communication, Ericsson Review No. 2, 2000
- [5] R. C. Jr., Adaptative Antennas: Concepts and Performance. Englewood Cliffs, NJ, Prentice Hall, 1988
- [6] J. G. Proakis, Digital Communications. New York, NY: McGraw Hill, 1995.
- [7] D. Bartolome, A. Perez-Neira, A. Pascual-Iserte, Blind and Semiblind Spatio-temporal Diversity for OFDM Systems Proceedings of ICASSP'02, Orlando, USA, May 2002

AUTORES



Marcel Palou nació en 1979 y cursó COU en Estados Unidos durante el curso académico 1996-1997. En 1997 inició los estudios de ingeniero de telecomunicación en la Escuela Técnica Superior de Ingeniería de Telecomunicación de Barcelona (ETSETB) de la Universitat Politècnica de Catalunya (UPC).

En 2002 se incorporó como becario a la empresa Tradia dentro de un programa-convenio de cooperación educativa con la UPC. Actualmente está en su última etapa de los estudios de ingeniero de telecomunicación llevando a cabo su proyecto final de carrera sobre técnicas de procesado de señal aplicadas a sistemas WLAN.



Antonio Pascual nació en 1977 en Barcelona. Cursó los estudios de ingeniero superior de telecomunicación en el período 1995-2000 en la Escuela Técnica Superior de Ingeniería de Telecomunicación de Barcelona (ETSETB) de la Universitat Politècnica de Catalunya (UPC), obteniendo el premio nacional de fin de carrera correspondiente al curso

2000-2001. Durante el período de septiembre de 1998 hasta junio de 1999 trabajó como becario en el departamento de ingeniería electrónica de la UPC, y desde junio de 1999, y hasta enero de 2001, trabajó, primero como becario y después como ingeniero contratado, en el departamento de I+D de Retevisión, involucrándose en temas relacionados con la implantación de las redes de televisión digital y radio digital terrestre en España. En enero de 2001 se incorporó como estudiante de doctorado y becario del programa de «Formació d'Investigadors» de la Generalitat de Catalunya al departamento de Teoría de la Señal y Comunicaciones (TSC) de la UPC, donde actualmente desarrolla en su última etapa de tesis actividades de investigación en el grupo de procesado de arrays y señales multicanal. Desde septiembre de 2003, es profesor en la Escuela Politécnica Superior de Castelldefels (EPSC) de la UPC.



DESARROLLO DE UN TERMINAL HIBRIDO GPRS-WLAN

Oscar García Alcoceba

RESUMEN

Internet está presente en prácticamente todos los lugares gracias al uso de los sistemas celulares como GPRS, UMTS, o redes de área local sin hilos (WLAN). Es lo que podemos considerar, de forma genérica, como Mobile Internet, pero debajo de esta denominación se esconden diferencias entre las diferentes formas de acceso. Mientras que unas, las WLAN, están optimizadas para dar servicio en interiores con baja movilidad las soluciones celulares atienden de forma satisfactoria los dominios públicos con movilidad. No parece que a corto plazo un tipo de acceso sin hilos se vaya a imponer sobre los otros. Más bien lo que tendremos es una complementariedad. Ante tal situación un usuario que quiere beneficiarse al máximo deberá disponer de un dispositivo móvil (PC portátil o PDA) que disponga de dos interfaces. Con esta solución podremos crear una sesión en una red y mantenerla mientras que sigamos bajo la cobertura de la misma. Si lo que queremos es mantener la sesión a pesar de nuestra movilidad entre redes debemos usar una solución de IP móvil. Con esta solución podremos efectuar descargas de ficheros o estar disponibles en una aplicación de mensajería instantánea sin ninguna interrupción a pesar de nuestra movilidad.

diferentes con IP móvil (figura 1). El software desarrollado debe trabajar de forma transparente al usuario ofreciendo la mejor conectividad en todo momento. El diseño realizado usa interfaz WLAN IEEE802.11b y GPRS, pero es perfectamente generalizable a otros tipos de WLANs o incluso a UMTS.

1 INTRODUCCIÓN

Gracias a la generalización de los accesos de banda ancha en la red fija (ADSL, Cable,...), se ha extendido el uso de nuevos servicios, que aprovechan la disponibilidad de mayor velocidad. Paralelamente, existe un creciente interés en que estos servicios estén disponibles en cualquier lugar y a través de cualquier terminal. Esta motivación es la que ha dado lugar al nacimiento de la Internet Móvil. Algunos estudios, como los del operador Telia [1], ponen de manifiesto esta tendencia y a la vez la imposibilidad de que una única tecnología pueda atender toda la demanda futura. En este mismo estudio se propone disponer de coberturas solapadas que permitan repartir los usuarios de acuerdo a sus requerimientos entre los diferentes sistemas radio. Atendiendo a esta situación futura es razonable pensar en terminales multisistema, capaces de no sólo de comunicarse usando diferentes banda de frecuencia, sino con diferentes protocolos. En los «Hot spots» o puntos de alta densidad de usuarios móviles ya se empiezan a adoptar esta solución descrita. La capacidad de los sistemas celulares GPRS o incluso UMTS puede llegar a resultar insuficiente, ofreciéndose cobertura complementaria con sistema WLAN. Esta situación tiene su reflejo también en los organismos de estandarización, como el 3GPP, que ya está trabajando para la integración a nivel de red fija de los sistemas de acceso basados en WLAN con los de UMTS.

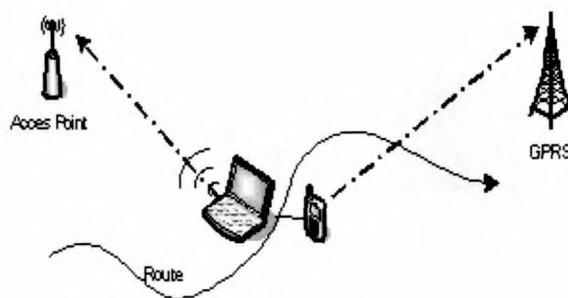


Figura 1.-Visión del concepto de terminal híbrido como aquel que dispone de dos interfaces y permite alternar la comunicación entre redes dependiendo de su disponibilidad

Considerando el entorno descrito previamente se planteó un trabajo de investigación consistente en el desarrollo de un software que permita a un ordenador funcionando con el sistema operativo Linux soportar movilidad entre redes

Disponer de terminales con dos o más interfaces se justifica también por otro interés bastante diferente al presentado en los párrafos anteriores. En la actualidad se dispone de un sistema de televisión digital terrestre (TDT) que permite la difusión de contenidos en práctica totalidad del territorio. Algunos fabricantes de equipos y operadores del servicios han visto la posibilidad de usar los canales de TDT para la difusión de tráfico IP, en lo que se denomina datacast [10]. Se puede difundir paquetes IP en modo multicast a terminales móviles y usar los canales ofrecidos tradicionalmente por los sistemas móviles como canales de retorno (figura 2). Entre las realizaciones de este concepto

destacar la descrita en [2], que a su vez esta basado en el proyecto europeo MEMO [8] (que utilizaba DAB- GSM) del cual incluso se realizó un prototipo de terminal. Todas estas propuestas se han encontrado con diversos problemas para su aplicación práctica en el mercado, la principal dificultad, es el hecho de que son fuertemente dependientes del éxito de las tecnologías DVB-T o DAB. Estas tecnologías están encontrando más dificultades de las esperadas debido al alto coste de los receptores, y la poca cuota de mercado existente hasta el momento. Para aliviar este problema se ha llegado a plantear el uso de repetidores de canales datacast que usen una interfaz WLAN para llegar a los usuarios. Esta propuesta consistente en sustituir el último tramo de la distribución de la señal procedente de DVB-T y hacerlo mediante WLAN incrementa el número de terminales que podrían recibir el flujo de paquetes IP.

Las soluciones datacast son interesantes para ofrecer aplicaciones de televisión interactiva, en la que un colectivo de usuarios está recibiendo un mismo flujo de información y que se precisa un canal de retorno para poder interactuar (pago de contenidos, respuesta a concursos, encuestas). Se puede observar que nuevamente se requiere el uso de dos interfaces, en este caso simultáneamente, para poder soportar la funcionalidad descrita.

Ante las posibilidades que ofrece un terminal móvil con dos interfaces se ha visto interesante realizar un prototipo de terminal híbrido que alternara la comunicación a través de los interfaces WLAN y GPRS, en función de los datos sobre el estado de los mismos. Esta idea no es del todo novedosa ya que otros proyectos como el descrito en [3] persiguen el mismo objetivo. En una primera aproximación se seleccionará el interfaz en función de su disponibilidad, pero en una

evolución futura se plantea el elegir el tipo de interfaz en función del tipo de datos o incluso la dirección de los mismos..

La realización del prototipo requiere profundizar en algunos aspectos clave a los que se dará mas detalle en las siguientes secciones. Se explicará las particularidades de la comunicación vía un MODEM GPRS, lo mismo para la comunicación y obtención de estadísticas de un interfaz WLAN. Por ultimo veremos algunos conceptos relacionados con Mobile IP. Esta solución nos permitirá mantener la misma IP durante toda la comunicación, evitando la perdida de la conexión al realizar un cambio entre redes.

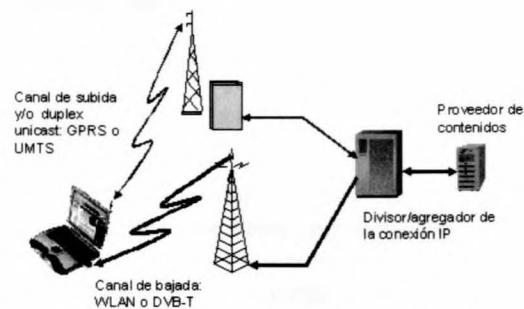


Figura 2.- Esquema de una solución datacast cambiando con un acceso móvil

2 COMUNICACIÓN VIA GPRS

La comunicación a través de un modem GPRS, para un usuario, puede parecer igual a la que tiene lugar cuando utilizamos un modem de 56K; aunque en realidad hay grandes diferencias. El modem GPRS establece con el terminal de datos una comunicación PPP, este adapta los

CAPAS OSI

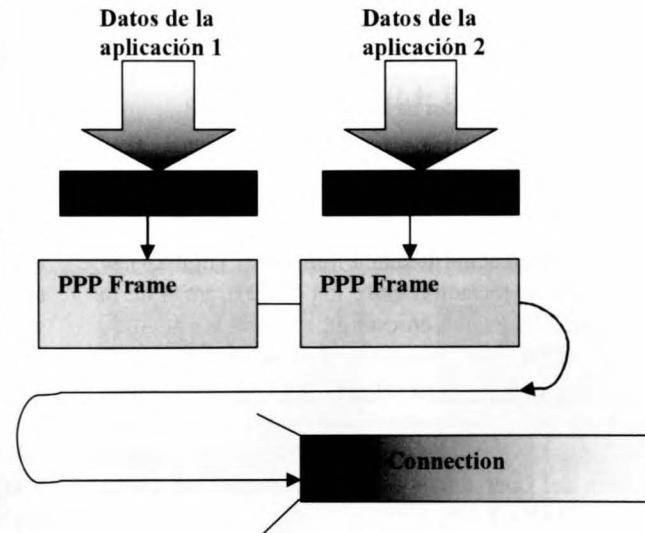
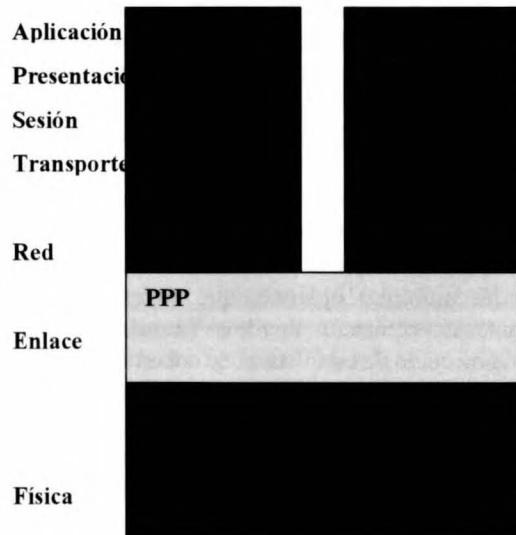


Figura 3.- Pila de protocolos PPP



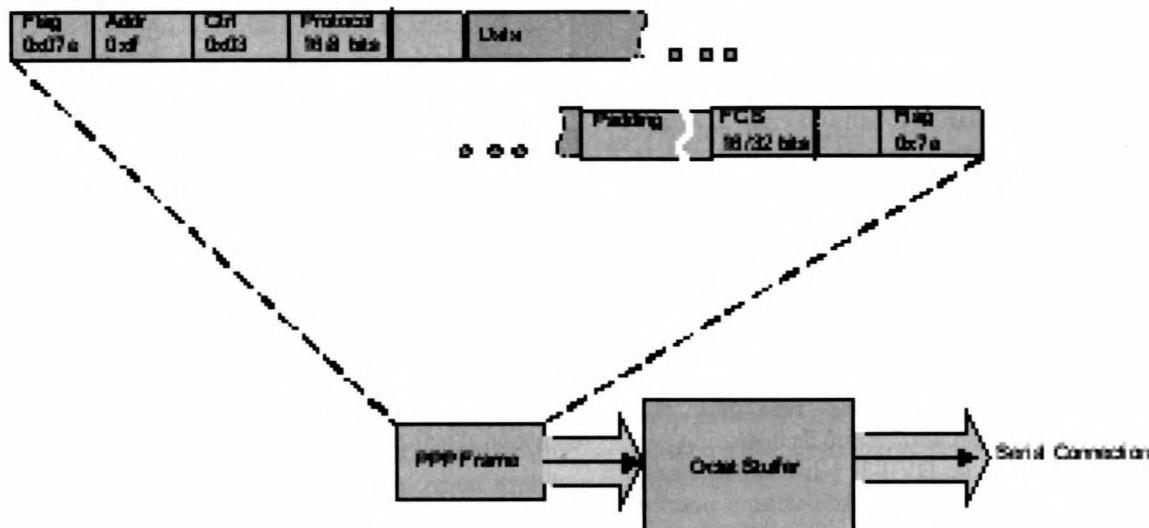


Figura 4.-Trama PPP

datos de la comunicación, a los protocolos y recursos GPRS, que será los que utilizará desde él hasta la red IP. Por el contrario en una comunicación a través de un modem de 56K la comunicación PPP se establece entre los dos terminales de datos extremos.

La finalidad del protocolo PPP, es adaptar los datos al puerto serie así como constituir el nivel de enlace de la pila de protocolos que se forma para esta comunicación. Los diferentes campos de la trama PPP, nos indicarán la longitud de esta, el tipo de paquete que contiene y nos permitirán verificar la integridad del paquete.

estados previo al inicio de la comunicación. En la figura 5, podemos ver el diagrama de estados por los que debemos pasar al establecer una comunicación PPP. Después de la posible autenticación se establecerá la negociación NCP (Network Control Protocol), de modo que se establece la negociación de los parámetros entre los extremos de la comunicación para los diferentes niveles de red.

En nuestro caso será importante que a la hora de negociar los parámetros del nivel IPCP, nuestro operador GPRS nos asigne una IP pública. Esto resultará fundamental para un correcto funcionamiento del cliente de Mobile IP, y nos marcará el tipo de acceso que ofrecerá nuestro proveedor GPRS, que será en modo transparente.

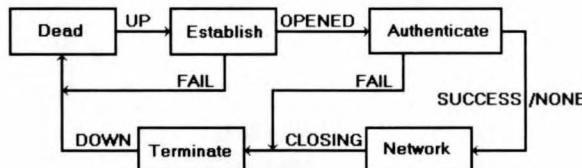


Figura 5.- Estados de la comunicación PPP

Previa a la comunicación mediante tramas PPP (figura 4), se establece una negociación entre los dos extremos de la comunicación PPP, que en este caso son el terminal de datos y el modem GPRS, mediante el protocolo LCP (Link Control Protocol), es aquí cuando podemos realizar el intercambio de los diferentes mensajes que servirán para autenticarnos. El protocolo utilizado para autenticarnos dependerá del operador, ya que disponemos de varias opciones CHAP o PAP.

Dentro de la negociación, pasamos por toda una serie de

desde el modem hasta la red IP pública nos moveremos en los protocolos GPRS (figura 6). Básicamente para establecer una comunicación GPRS, necesitaremos tener activado un contexto PDP. El contexto PDP indicará entre otras cosas el tipo de datos que enviaremos, el punto de salida de la red GPRS o APN, así como otros parámetros que podemos fijar, como puede ser la QoS, etc., dependiendo de las funciones admite el operador.

El diálogo del terminal con el modem GPRS se realiza mediante una extensión de los conocidos comandos Hayes. Entre las múltiples opciones que podemos controlar mediante estos comandos, una de las que más nos interesaría será la obtención de estadísticas de cobertura, para poder realizar con estos datos las decisiones de encaminamiento, de forma similar a como se propone en [3]. Dentro de los parámetros que podemos obtener será importante que nos basemos en la BER (Bit Error Rate) calculada a través de la RXQUAL, más que en RSSI. Hay múltiples opciones más que se pueden controlar a través de los comandos Hayes desde obtener el nivel de carga de la batería, la identidad de la celda a la que estamos conectados, etc.

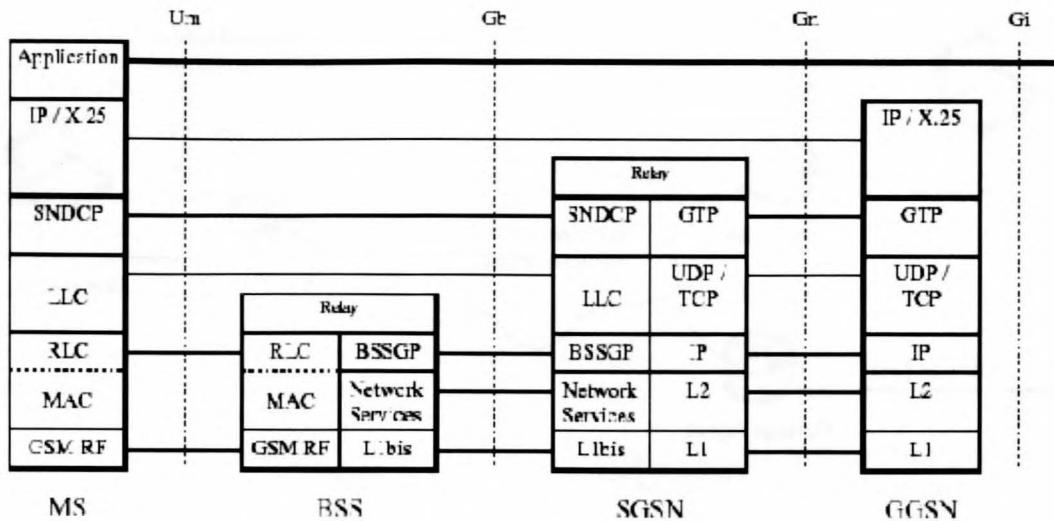


Figura 6.- Pila de protocolos GPRS

3 RED WLAN

La comunicación a través de la tarjeta wireless 802.11b, en general no presenta demasiadas complicaciones a nivel de funcionamiento, más que instalar o encontrar el driver adecuado para hacer funcionar y controlar la tarjeta WLAN en Linux. Una vez instalada, la comunicación a través de la tarjeta WLAN usa la misma pila de protocolos que con una tarjeta Ethernet. Del trabajo realizado con la WLAN lo mas relevante es el método utilizado para recoger las estadísticas de cobertura y estado del enlace de la tarjeta. Esta tarea resultará complicada puesto que cada fabricante utiliza un método diferente para realizar esta función, y por tanto hay que utilizar estructuras de datos diferentes dependiendo de la identidad de la tarjeta.

Ante la heterogeneidad de métodos que utilizaban los diversos fabricantes en sus drivers para recoger estadísticas, se impuso la necesidad de definir una especie de API Wireless. Esta tenía por finalidad permitir al usuario manipular los dispositivos de red de una forma estándar y uniforme, sin importar el fabricante de la tarjeta, o el interfaz de comunicación de esta. Esta estandarización afecta a los métodos utilizados, ya que debido a la diferente naturaleza de cada tarjeta los valores serán diferentes. Necesitamos que esta interfaz sea flexible a la par que extensible, para poder ir adaptando diferentes fabricantes.. La principal necesidad que se busca cubrir con esta herramienta era la configuración de dispositivos, aunque también interesaba la recogida de estadísticas y las posibilidades que abría para el desarrollo de aplicaciones que puedan aprovecharse de las capacidades wireless. Para conseguir esto, se debían realizar una serie de modificaciones, tanto en el sistema operativo, como en los drivers de los diferentes dispositivos. Las variaciones en el sistema no suponían más que una pequeña modificación o extensión de la pila de red de Linux, por lo cual se decidió

darle el nombre de «Wireless Extensions», el impulsor de las cuales es Jean Tourrilhes.

La herramienta «Wireless extensions», tendrá definidos tres 3 niveles de ejecución o abstracción diferentes. Desde el nivel más externo al más cercano al hardware tenemos, en primer lugar, el interfaz de usuario, que es un conjunto de herramientas que nos permiten las extensiones; la segunda parte es la modificación del kernel de Linux para definir y soportar las extensiones, y por ultimo el interfaz hardware y su implementación en cada driver de red, con el que se mapean las extensiones a las opciones disponibles por el hardware

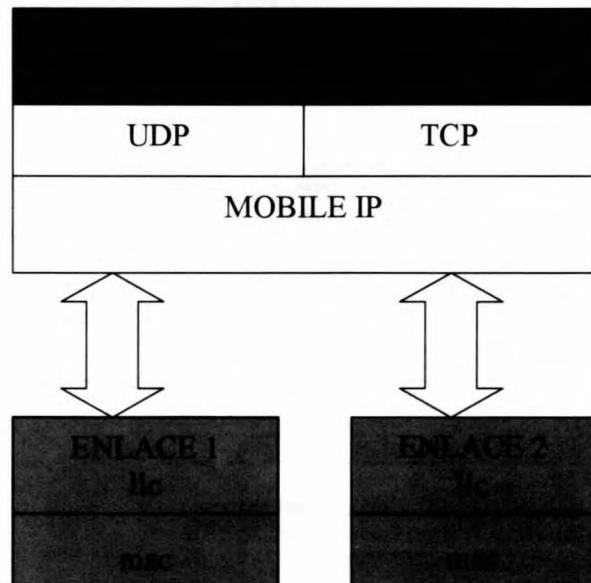


Figura 7.- Pila Mobile IP

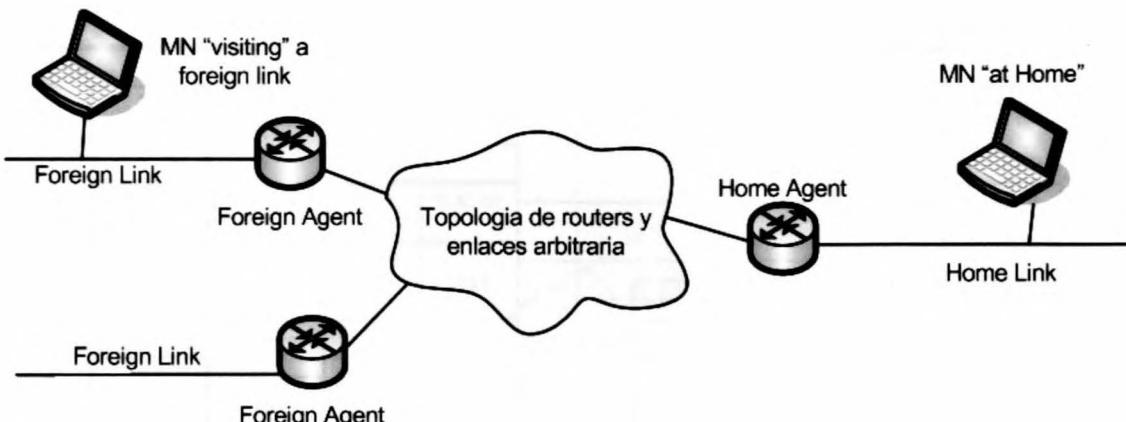


Figura 8.- Esquema de comunicación Mobile IP

4 MOBILE IP

De la necesidad de poder mantener la comunicación, aun cambiando de punto de conexión, surgió un grupo de trabajo dentro de IETF, el *IP Routing for Wireless/Mobile Host working group* que fue el que origino el estándar Mobile IP.

Mobile IP es una de los protocolos definidos por el IETF para Internet. Los dos objetivos fundamentales de este protocolo son, por un lado garantizar la continuidad de la comunicación aún cuando estemos cambiando de red, y por otro lado mantener la disponibilidad del equipo para el resto de equipos de la red. Esto hace que sea un protocolo independiente del nivel de enlace, de manera que nos permitirá cambiar nuestro punto de conexión a la red, manteniendo la misma dirección IP.

En el fondo el estandar Mobile IP no es más que una extensión del protocolo IP (figura 7), adaptandolo a las nuevas necesidades de movilidad. Devido a las condiciones de diseño del protocolo Ipv4, deberemos añadir tres entidades funcionales en nuestra red, para proporcionar las funciones del estandar Mobile IP. En la figura 8 podemos ver representadas las diferentes entidades, así como la estructura de una red que implementa las funciones de Movilidad.

Mobile IP define tres entidades funcionales donde tienen que estar implementados los protocolos de movilidad. El *Mobile Node(MN)*, será el nodo que cambia de punto de acceso a la red, manteniendo sus comunicaciones de salida y usando solo su dirección home permanente para ello. Esto es precisamente una de las funcionalidades que buscamos en nuestro terminal híbrido.

El *Home Agent(HA)*, será un enrutador con una interfaz en el home link, el cual realizará diversas funciones: estar informado de la localización del mobile node y su dirección en la red local, interceptar los paquetes destinados al MN y entunelarlos hasta su localización actual.

El *Foreign Agent(FA)*, suele ser un enrutador en la red visitada (foreign link). Esta es la única entidad que es opcional, su presencia depende en gran medida, del modo de funcionamiento del sistema de movilidad.

El sistema tiene dos modos de funcionamiento fundamentales. El modo FADecapsulation, en el que al crear el tunel entre la red local y la visitada, para redireccionar los paquetes, este finaliza en el FA, y los paquetes son reenviados por el FA al MN que estará identificado en una tabla interna del FA. Y el modo MNDecapsulation, en el cual el tunel finaliza directamente en el MN, en este modo

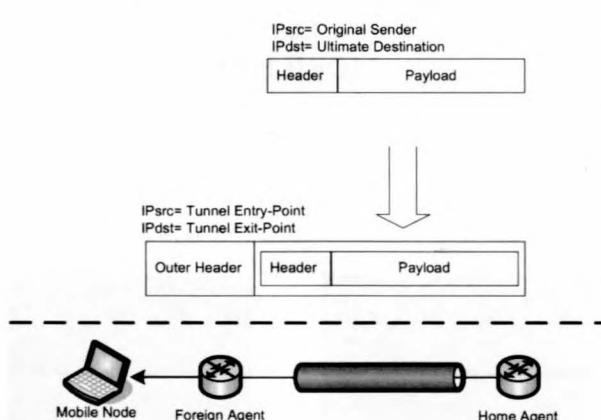


Figura 9.- Entunelado FADecapsulation

será el HA el que tendrá identificado directamente en su tabla al MN. En este último modo el FA, no suele utilizarse.

Existen diferentes alternativas de funcionamiento o modos, además del original definido en el primer RFC, han aparecido muchos más en los siguientes RFCs. Un ejemplo de esta evolución la tenemos en la definición de Mobile IPv6, realizada a la vez que se definían las características IPv6. Al contrario de lo que pasa en IPv4, en el que hemos de definir estas entidades funcionales, en IPv6 el estandar Mobile IP ya viene integrado en los mecanismos y campos asignados a este protocolo.

El modo original de funcionamiento, era FADecapsulation con Triangle Tunneling (figura 9); de modo que la comunicación del MN al Host de destino se realizaba en claro por el mismo camino que lo haría en una comunicación normal, en cambio al responder el Host al MN, estos paquetes irán el Home Link donde serán recogidos por el HA y entunelados hasta el FA que los extraerá del tunel y los entregará al MN.

Entre los diferentes modos definidos a posteriori, el modo MNDecapsulation Reverse Tunneling, que tal vez sea el más interesante para el desarrollo de nuestro terminal. En este modo la información va entunelada hasta el MN, que además envía sus paquetes por el mismo túnel y es el HA quien los desentunela y envía a su destino. Para que este modo sea posible necesitamos que el MN tenga una IP pública.

Por ultimo, es interesante destacar que, la IP móvil nos aporta los beneficios ya citados, pero tiene algunos inconvenientes que debemos tener en cuenta. Una de las principales desventajas, en nuestro caso, será que, nos limitará la velocidad efectiva de comunicación debido al encapsulado de los paquetes IP, que atraviesan el tunel, lo que penalizará especialmente la comunicación a través de la conexión más lenta, ya que verá reducida aún más su velocidad.

5 CONCLUSIONES

Finalmente después de ver el funcionamiento de nuestro prototipo de terminal, podemos obtener toda una serie de observaciones y conclusiones, que nos pueden ser de utilidad en un futuro.

El principal defecto que tiene el prototipo será la comunicación vía GPRS, debido a la lentitud y los retardos que presenta. Esto es debido al entunelado en modo reverso, que hace que todos los paquetes que circulan por el enlace, tanto en subida como en bajada, vayan entunelados , se podría solucionar si en lugar de desentunelar en terminal lo hiciéramos en el FA , utilizando el modo FADecapsulation de modo que por el canal radio viajaran los paquetes sin encapsular. Para lo cual necesitaríamos modificar la infraestructura del operador

de telefonía móvil para que proporcionara Mobile IP, tal y como prevé el propio estándar; otra solución sería usar el Triangle tunneling de esta manera los paquetes de subida no irían encapsulados aunque esto puede provocarnos otros problemas con el enrutamiento en la red. Un tema también interesante podría ser adaptar el diseño a otra tecnología que nos ofrezca un canal de comunicaciones con un ancho de banda mayor, este podría ser el caso de la tecnología UMTS.

REFERENCIAS

- [1] Karlsson, Peter; *Integration of WLAN and Cellular Networks*. Telia Research AB. Malmö, Suecia.
- [2] Rauch, Christian; Kellerer, Wolfgang; Sties, Peter; *Hybrid Mobile Interactive Services combining DVB-T and GPRS*.
- [3] EURESCOM P1013 FIT-MIP. *Test objectives, scenarios and results .Part 2: GPRS-WLAN handover and TCP*
- [4] Stevens, W. Richard; Wright, Gary R.; *TCP/IP Illustrated, Volume 2: The implementation*; Editorial Addison-Wesley professional computing series. Febrero 2002.
- [5] Rubini, Alessandro; Corbet, Jonathan; *LINUX Device drivers. 2nd edition*; Editorial O'Reilly, 2002.
- [6] Sun, Andrew; *Using and Managing PPP*; Editorial O'Reilly. Sebastopol, marzo del 1999.
- [7] Salomon, James D.; *Mobile IP: the internet unplugged*; Editorial Prentice Hall, 1998.
- [8] <http://MEMO.lboro.ac.uk>
- [9] www.ietf.org
- [10] Paila, T.; *Mobile Internet over IP data broadcast*; Telecommunications, 2003. ICT 2003. 10th International Conference on , Volumen: 1 , 23 de Febrero al 1 de Marzo de 2003 , paginas:19 - 24 vol.1

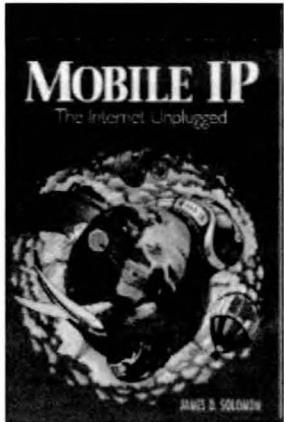
AUTOR

Oscar García Alcoleba



Ha estudiado en el ETSETB y actualmente está finalizando el PFC «Estudio de implementación de un sistema híbrido WLAN, GPRS» en el departamento de Telemática.





REDES CELULARES 4G BASADAS EN MOBILE IPV6 CON SOPORTE DE NODOS DURMIENTES

Sara Berzosa Calpe (proyectista) y Rafael Vidal Ferrer

Sara.Berzosa@estudiante.upc.es: Estudiante de segundo ciclo de Telcomunicaciones en Castelldefels

Universitat Politècnica de Catalunya

rafael.vidal@entel.upc.es: Departamento de Ingeniería Telemática

Grupo de redes inalámbricas

Universitat Politècnica de Catalunya

Abstract. – Este artículo describe el protocolo Mobile IPv6 que permite la movilidad de nodos entre redes sin pérdida de conectividad y que podría ser uno de los pilares de la denominada 4G de redes celulares. Además se describe las modificaciones teóricas y prácticas, a partir de una implementación en código abierto de Mobile IPv6, para que este protocolo soporte de nodos móviles durmientes, nodos que para ahorrar batería desactivan periódicamente su interfaz radio cuando no la necesitan. Todo ello probado sobre un maqueta IPv6.

1. INTRODUCCIÓN

Parece claro que la denominada cuarta generación de redes celulares (4G) tendrá como ejes estratégicos la utilización de múltiples interfaces radio y la utilización de IP como protocolo de transporte de datos y señalización. Estas dos tendencias ya son claramente observables en la actualidad. IP se ha convertido en el protocolo de interconexión universal y las propuestas de redes de 3G del 3GPP y 3GPP2 muestran una clara evolución hacia redes todo IP (All-IP) [1]. En el caso de 3GPP, se da un paso más con la adopción de IPv6 como protocolo de red en lugar de IPv4 [2]. IPv6 [3] pone solución a la actual escasez de direcciones, además de, entre otras cosas, ofrecer integrar de serie mecanismos de seguridad y un mejor soporte de la movilidad. Es por todo ello que se espera que sea el protocolo de red de la 4G de redes celulares.

Por otro lado, los nodos móviles pueden disponer de más de una interfaz (UMTS, GPRS, WLAN o Bluetooth) de manera que el usuario o incluso el propio terminal pueden decidir cual de las interfaces disponibles es la más adecuada en cada momento. El cambio de red de acceso suele conllevar un cambio de dirección IP en el nodo móvil lo que supone la caída de todas sus comunicaciones en curso. Como solución a este problema, el IETF estandarizó el protocolo Mobile IP (MIP) [4] que permite el cambio de subred e IP asociada de manera transparente al usuario, es decir manteniendo sus conexiones activas. Ampliamente soportado por fabricantes

como Cisco, sobretodo para el mercado de las redes WLANs, MIP forma parte del estándar Wireless IP [5] de la propuesta de 3G cdma2000 del 3GPP2, y en su momento, también fue estudiada su utilización por el 3GPP en futuras versiones de UMTS. MIP abre la puerta a la utilización de IP no solo para el transporte sino también para el soporte de la movilidad. Para que ello sea posible se está estudiando la conveniencia de soportar otras funciones de las redes celulares como la que es objeto de estudio en este artículo: el soporte de los nodos durmientes.

Es habitual en redes celulares que los nodos móviles que no tienen una comunicación en curso puedan pasar a un estado denominado durmiente en el que reducen su monitorización del medio radio con el objetivo de ahorrar baterías. Con el mismo motivo, en este estado el nodo móvil en lugar de informar a la red de cada cambio de celda que realiza se limita a informar solo cuando cambia de un grupo predefinido de celdas, denominado área de localización, a otro. Esto conlleva que la red soporte un método de búsqueda, también llamado paging, que permita averiguar exactamente en qué celda se encuentra un nodo durmiente para poder avisarle, por ejemplo, de que tiene una llamada.

Este artículo aborda la primera fase de un trabajo que tiene como objetivo final el disponer de una maqueta de red 4G basada en Mobile IPv6 (MIPv6) [6] y con soporte de nodos durmientes y paging a nivel IP. MIPv6 es la versión para IPv6 de MIP. El hecho de trabajar con IPv6 permite a MIPv6 respecto a MIP ganar en eficiencia e incluso reduciendo el número de nodos especiales necesarios para su despliegue. Todo ello se explica en el apartado 2 de este artículo donde se describe el protocolo MIPv6. En el siguiente apartado se describe el concepto y la propuesta utilizada para el soporte de nodos durmientes y paging IP. Para añadir estas funcionalidades a MIPv6 se ha estudiado una implementación de código abierto del protocolo descrita en el apartado 4 que permita su modificación. Para probar esta implementación y las posteriores modificaciones descritas en el apartado 5 se ha construido una maqueta explicada

en el apartado 6.

2.MOBILE IPV6

En una comunicación MIPv6 intervienen principalmente tres elementos: *Mobile Node* (MN), *Home Agent* (HA) y *Correspondent Node* (CN). Un MN es un nodo que cambia su punto de conexión a Internet. Un Home Agent es el agente con el cual el MN registra sus direcciones. Este agente es el encargado de interceptar y reenviar los paquetes dirigidos al MN mientras éste está en otra subred. Finalmente, un CN es un nodo que establece comunicación con un MN. Los CNs también pueden ser móviles.

En MIPv6, los MNs se identifican siempre mediante su *home address* (HoA) en lugar de identificarse mediante su punto de conexión a Internet. La HoA es una dirección IP asignada a un MN dentro de su *home network*. Mientras un MN está en su *home network*, los paquetes dirigidos a su HoA se encaminan hacia esa red usando los mecanismos de encaminamiento convencionales de Internet.

Un nodo detecta que ha cambiado de red mediante la recepción de los anuncios de router (*Router Advertisements*, RA), en los cuales se difunden los prefijos de las redes. Cuando un MN se conecta a una *foreign network*, también estará alcanzable gracias a una o más *care-of addresses* (CoA). Una CoA es una dirección IP asociada a un MN que está fuera de su *home network*, formada con el prefijo de la nueva red en la que se encuentra. El MN puede adquirir su CoA mediante mecanismos convencionales de IPv6 como la autoconfiguración *stateless* (gracias a los anuncios de prefijo de los routers) o *stateful* (gracias al DHCP). Mientras el MN esté en la *foreign network*, los paquetes dirigidos

a su CoA serán encaminados hacia el MN.

La asociación entre la HoA de un MN y su CoA es lo que se conoce con el nombre de *binding*. Cuando un MN cambia de red, registra su CoA primaria con el HA. El MN realiza el registro de esta *binding* enviando un mensaje *Binding Update* (1) al HA. El HA responde al MN retornando un mensaje *Binding Acknowledgement* (2). La siguiente figura muestra el proceso: Cuando se habla de un CN, se hace referencia a cualquier nodo con el que se comunica un MN. Un CN puede ser un nodo estático o móvil. Los MNs pueden proporcionar información a los CN sobre su localización actual mediante un registro. Como parte de este procedimiento, se realiza un test para autorizar el establecimiento de la *binding* llamado *return routability test* para autorizar el establecimiento de la *binding*.

Entre un MN y un CN existen dos modos de comunicación. El primer modo, el *tunneling bidireccional*, no requiere soporte de Mobile IPv6 en el CN, y funciona incluso si el MN no ha registrado su binding actual con el CN. La figura 2 muestra el proceso. Cuando el CN quiere enviar paquetes al MN, los envía a su HoA (1), y el HA se encarga de interceptarlos y entunelarlos hacia el MN (2). Los paquetes dirigidos al CN se entunelan desde el MN hacia el HA (*reverse tunneling*) (3) y éste los encamina desde la *home network* hacia el CN (4). En este modo, el HA usa proxy *Neighbor Discovery* para interceptar los paquetes IPv6 dirigidos a la HoA del MN. Cada paquete interceptado se entunela a la CoA primaria del MN. Este entunelado se realiza

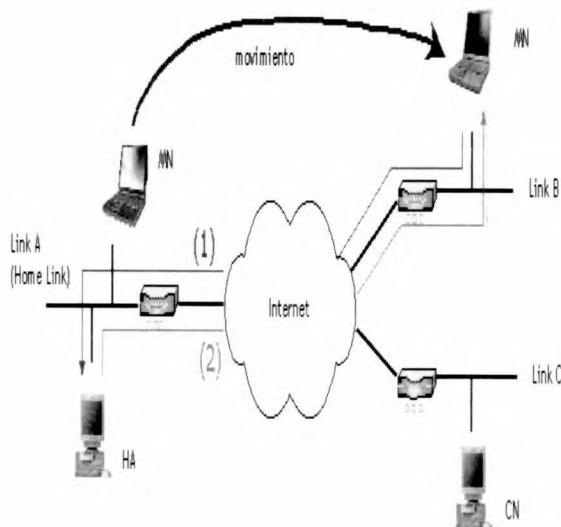


Figura 1. Proceso de registro con el HA

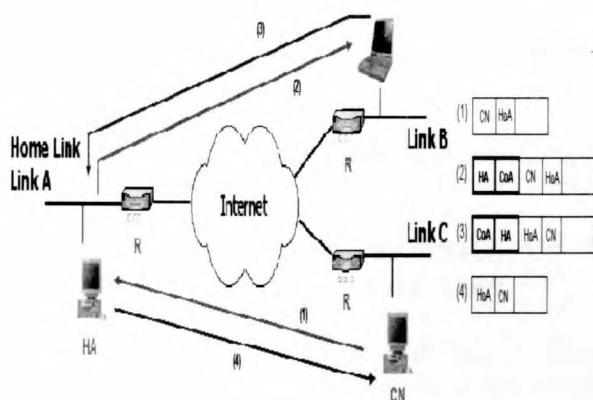


Figura 2. Tunneling bidireccional

usando encapsulado IPv6 [7].

La figura 3 muestra el segundo modo, llamado *route optimization*. Éste requiere que el MN registre su *binding* actual con el CN. En este modo, los paquetes enviados del CN al MN se encaminan directamente a la CoA del MN (1). Cuando el CN envía un paquete a cualquier destino IPv6, comprueba en las *bindings* de su caché si existe una entrada con la dirección destino del paquete. Si encuentra la entrada, el CN usa una extensión de cabecera del tipo *routing*



header para encaminar el paquete al MN (2). El hecho de encaminar los paquetes directamente hacia la CoA del MN permite usar el camino más corto para la comunicación. También elimina congestión en el HA del MN y en la *home network*. Además, se reduce el impacto de posibles fallos tanto en el HA como en la *home network*.

Cuando los paquetes se encaminan directamente hacia el MN, el CN establece la *Destination Address* de la cabecera IPv6 a la CoA del MN. Un nuevo tipo de cabecera de encaminamiento se añade para transportar la HoA. De manera similar, el MN establece la *Source Address* en la cabecera IPv6 a su CoA actual, y añade una nueva *destination option* llamada *Home Address* para transportar su HoA. La inclusión de las HoA en estos paquetes hace el que el uso de la CoA sea transparente para las capas

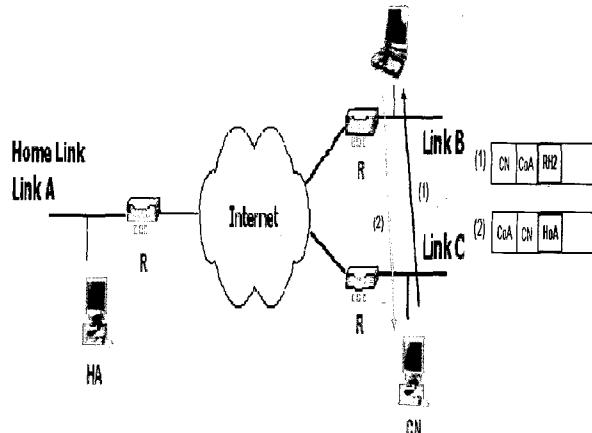


Figura 3. Route optimization

superiores a la de red.

3.PAGING IP: ARQUITECTURA Y PROTOCOLO UTILIZADO

La necesidad de un protocolo de paging a nivel IP fue estudiada por el IETF en el RFC 3132 [8] con el nombre de *Dormant Mode Host Alerting* (DMHA). Este trabajo fue acompañado por el RFC 3154 [9], una guía completa de los requisitos y la arquitectura funcional que debe seguir una solución de paging IP. Sin embargo este punto de partida no se tradujo en un protocolo de consenso [10].

A continuación se comentará la arquitectura descrita en el RFC 3154 y seguidamente el protocolo que se ha tomado como referencia para incorporarlo en el código de MIPv6.

3.1.Arquitectura de paging IP

El RFC 3154 define una arquitectura funcional compuesta 4 entidades. Son: el Host o MN, el *Tracking Agent* (TA), el *Dormant Monitoring Agent* (DMA) y el *Paging Agent* (PA). El TA es el encargado de controlar el estado del Host, activo o durmiente, y su localización. Esta localización se corresponde con un área de paging, formada por un conjunto de routers de acceso (*Access Routers*, ARs) que difunden un mismo identificador. El Host informa al TA de sus cambios de estado y área de paging. El PA es el encargado de mandar los mensajes de paging a Hosts durmientes previa consulta del TA. También es el encargado de la difusión de los identificadores de área de paging (*Paging Area Advertisements*, PAI). La recepción de estos identificadores permite al Host saber cuando cambia de área de paging y portanto avisar al TA. Finalmente, el DMA detecta la llegada de paquetes dirigidos a Hosts durmientes e indica al PA que debe buscar un Host. Cuando este pasa estado activo le

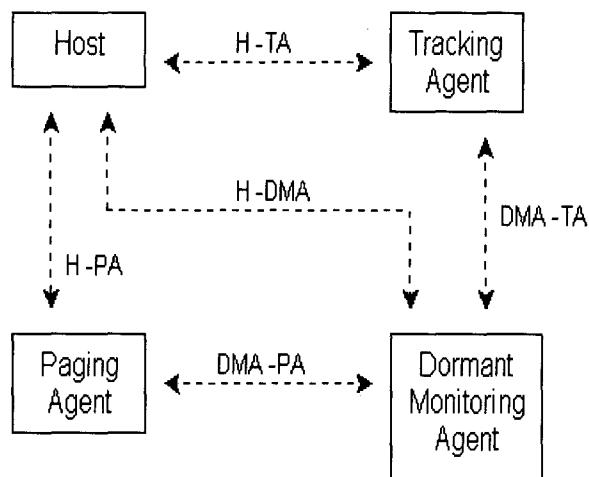


Figura 4. Arquitectura funcional paging IP

entrega los paquetes.

Por simplicidad supondremos que las funciones de TA y DMA están también incluidas en el PA y nos situaremos en un escenario de ejemplo como el de la Figura 5. En ella podemos observar dos áreas de paging, una formada por AR1 y AR2, y otra por AR3 y AR4. Cuando el host detecte un cambio de PAI informará al PA y cuando este reciba un paquete dirigido al Host le enviará un paquete de búsqueda

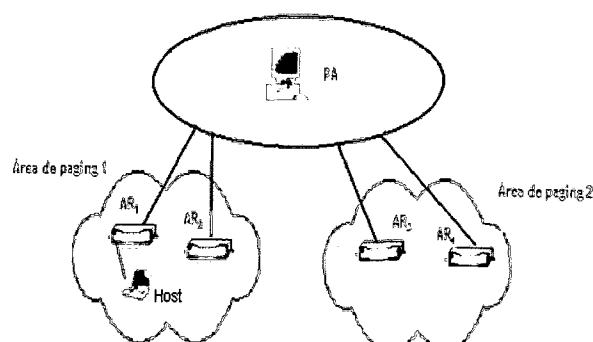


Figura 5. Escenario MIPv6 con paging IP

a todos los ARs asociados al PAI.

3.2. Protocolo paging IP utilizado

Siguiendo la arquitectura y el escenario descritos, la propuesta escogida [11] pretende integrar el soporte de nodos durmientes y paging IP a MIPv6. Así pues el Host de la Figura 5 se convertirá en un MN y aparecerá el HA. En este documento a un nodo durmiente se le asigna el estado denominado *idle*.

Otra de sus características es el uso del modo de registro explícito (*Explicit Idle State Registration*), que consiste en el envío por parte del MN de un mensaje *Idle State Request* (1) al PA para indicarle su paso a nodo durmiente e informarle de la área de paging en la que se encuentra. El PA cuando recibe este mensaje, debe actualizar internamente la información sobre el MN y confirmar la recepción del mensaje mediante un *Idle State Reply* (2). El MN entonces deberá enviar una *Binding Update* al HA (3), mediante la cual registrará la dirección del PA. El HA confirmará la recepción de la BU enviando una *Binding Acknowledgement* al MN (4). Si cualquiera de los dos mensajes de confirmación indican algún fallo en el registro, el MN debe permanecer en estado activo y registrar su CoA actual con el HA.

En la *Binding Update* al HA se registra la dirección del PA para mantener el protocolo de paging independiente de la entidad HA. Recordemos, que cuando el MN está fuera de la *home network*, el HA intercepta los paquetes dirigidos a la HoA del MN, para posteriormente enviárselos a su CoA. Así pues, cuando ahora el HA intercepte los paquetes dirigidos al MN (5), los enviará a la dirección del PA (6). Éste, cuando reciba los paquetes empezará el proceso de paging, y además tendrá que encargarse de guardar estos paquetes. De esta manera el registro del modo durmiente y el proceso de paging es transparente al HA. La siguiente figura muestra un esquema del proceso.

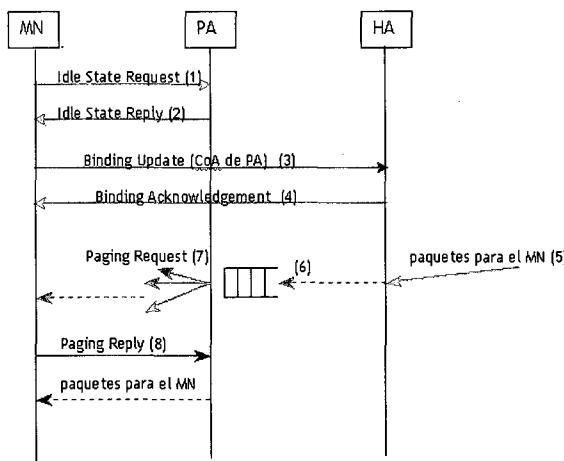


Figura 6. Proceso de registro y paging

Una vez el MN está en estado durmiente, puede pasar al estado activo cuando hay paquetes dirigidos a él y se realiza el proceso de paging o bien cuando decide volver a entrar en modo activo.

En el proceso de paging, el PA interroga simultáneamente con un mensaje *Paging Request* (7) a los ARs del área de paging en la que se encuentra el MN. Este mensaje ha de ser capaz de identificar a un nodo móvil en concreto: se consigue incluyendo un identificador único en el mensaje, como puede ser por ejemplo la HoA del MN. Cuando el MN recibe este mensaje, confirma que ha entrado en estado activo con un *Paging Reply* (8). Como *Paging Reply* se puede enviar una *Binding Update* (normal) al HA o cualquiera de los mensajes que servían para entrar en modo activo explicados en el párrafo anterior.

Cuando un MN cambia de área de paging, ha de realizar una actualización de localización. Esta actualización se lleva a cabo mediante el mensaje *Idle State Request*, indicando el identificador de área de paging. El PA ha de confirmar la recepción de esta actualización con un mensaje *Idle State Reply*. Si además de cambiar de área de paging se cambia también de subred, debe realizarse el correspondiente aviso de cambio de CoA.

Gracias al paging, el nodo móvil en lugar de informar de cada movimiento de subred que realiza, únicamente informa cuando cambia de área de localización, lo que comporta una clara disminución de señalización y el consiguiente ahorro de batería.

4. IMPLEMENTACIÓN DE MOBILE IPV6

En el proyecto se montó una maqueta con los protocolos IPv6 y MIPv6 para poder posteriormente probar las modificaciones de código realizadas para dar soporte de paging. Se usó una implementación de la especificación de MIPv6 para Linux creada por la *Helsinki University of Technology* (HUT): *MIPL Mobile IPv6 versión 1.0*. Esta implementación se instala como un parche para el kernel. La versión de MIPL Mobile IPv6 utilizada en este proyecto es la 1.0, funciona bajo el kernel 2.4.22 y sigue las especificaciones del draft [6].

4.1. Módulos MIPL Mobile IPv6 v1.0 y IPv6

Tanto el protocolo IPv6 como la versión del protocolo Mobile IPv6 para Linux de HUT están implementados como módulos de kernel.

Los módulos son trozos de código que se pueden cargar y descargar en el kernel bajo demanda, extendiendo su la funcionalidad del kernel base sin la necesidad de implementar las nuevas



funcionalidades directamente en él. Ofrecen varias ventajas; por una parte no hay que recompilar el kernel entero cada vez que se añade una nueva funcionalidad, lo que comporta un ahorro de tiempo y disminuye la posibilidad de introducir errores recompilando y reinstalando el kernel base. Además, el tamaño del kernel base no aumenta, por tanto se ahorra memoria, porque los módulos sólo se cargan cuando se van a usar.

El módulo MIPv6 se configura en el archivo `/etc/sysconfig/network-mip6.conf`. El script de inicio lee este archivo y carga unos módulos u otros dependiendo de la funcionalidad especificada, y los parámetros especificados en él se envían al módulo MIPv6 a través del proc-filesystem.

El script de inicio puede cargar uno o dos módulos; uno, si la funcionalidad es de CN, y dos, si la funcionalidad es de MN o HA, ya que en este caso carga el módulo que toca y además el módulo de CN.

Debido a que los módulo MIPv6 se insertan y se retiran del kernel de manera dinámica, no se pueden usar llamadas directas desde el kernel a los módulos MIPv6. En lugar de eso, las funciones de MIPv6 que pueden ser llamadas se definen en el código de IPv6 (en `mipglue.c`). En `mipglue.c` se comprueba si el puntero a la función está asignado, y si es así, se llama a la función. Los punteros a las funciones las asigna el módulo MIPv6 cuando se carga en el kernel. La ventaja de esto, como se ha comentado anteriormente, es que no se necesita compilar el kernel entero ni reiniciar el sistema cada vez que se quiere testear el módulo.

La figura 7 muestra un esquema de cómo se realizan las llamadas a funciones. El módulo `mipglue` actúa reenviando las llamadas que le llegan del módulo IPv6 hacia el módulo MIPv6 y viceversa. El funcionamiento de MIPv6 es transparente para el usuario y para las capas superiores a IP (transporte y aplicación). La movilidad transparente se consigue añadiendo al kernel el módulo MIPv6 y modificando el módulo IPv6 existente. El soporte para la movilidad se implementa añadiendo nuevas cabeceras de extensión al protocolo IPv6. Las tres entidades (HA, MN, CN) se comunican entre ellos la información relacionada con la movilidad a través de estas cabeceras de extensión.

Módulos y submódulos de MIPv6

La implementación de MIPv6 consta de cuatro módulos: `module_ha.c` (módulo HA), `module_mn.c` (módulo MN), `module_cn.c` (módulo CN) e `ipv6_tunnel.c` (módulo de tunneling IPv6-IPv6). Estos archivos contienen básicamente las funciones de inicialización. Además, en cada inicialización de módulo se establecen los punteros a las funciones del código de IPv6 que pueden llamarse desde Mobile IPv6 y se registran las rutinas que pueden ser llamadas cuando ocurre un evento. También es aquí donde se inician los demás submódulos. El módulo de tunneling maneja el encapsulado/desencapsulado IPv6-IPv6. La función de encapsulado la invoca el HA cuando intercepta un paquete que va destinado al nodo móvil y éste está fuera de su home network. El encapsulado añade la cabecera IPv6 para el túnel antes de la cabecera IPv6 existente.

Si se teclea el comando `lsmod` mientras se está ejecutando Mobile IPv6, se pueden observar los

Module	Size	Used by	Not tainted
<code>mip6_mn</code>	7148	0	(unused)
<code>ipv6_tunnel</code>	15192	1	[<code>mip6_mn</code>]
<code>mip6_base</code>	45880	0	[<code>mip6mn</code>]
<code>ipv6</code>	231924	-1	[<code>mip6_mn ipv6_tunnel mip6_base</code>]

módulos que hay cargados en el kernel:

Este ejemplo es de un MN. Los módulos cargados son `mip6_base` (módulo común a todas las entidades, el de CN), `mip6_mn` (módulo del MN) y `ipv6_tunnel` (módulo de túnel). El otro módulo cargado es el de `ipv6`. El comando `lsmod` también informa de otros parámetros, como el tamaño que ocupa el módulo, si se está usando o no, y si se está usando por quién (es decir, las dependencias). El -1 en la columna «used» para `ipv6` indica que este módulo no se puede descargar, ya que todavía está en desarrollo y descargarlo podría producir inestabilidades en el

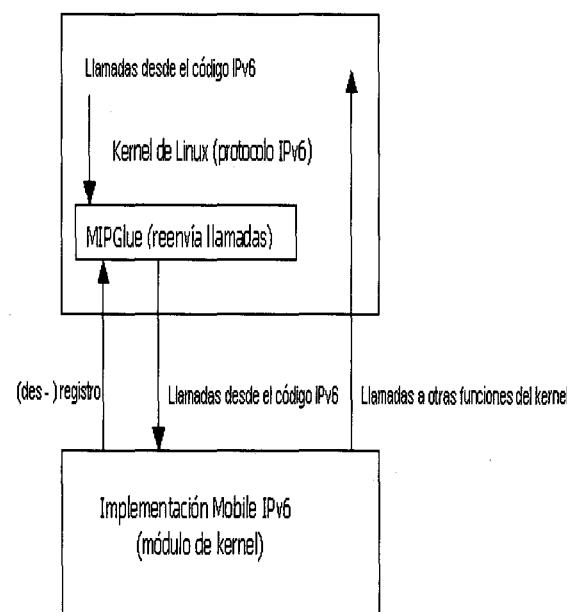


Figura 7. Esquema de comunicación entre IPv6, MIPv6 y el kernel

sistema.

Por otra parte, la implementación de Mobile IPv6 consta de diferentes submódulos. El concepto de submódulo no es el mismo que el de módulo. Cuando se habla de submódulo se hace referencia a una parte de código que realiza una serie de ejecuciones.

Módulo IPv6

Cuando el módulo IPv6 se inicia, lo primero que se hace es iniciar los submódulos *icmpv6*, *ndisc*, *igmpv6* y *ip6_tunnel*. Estos, a su vez, realizan un registro, creación e inicialización de los sockets de control *icmpv6*, *ndisc* e *igmpv6*. Posteriormente, inicia otros submódulos propiamente del protocolo IPv6, como *ip6_route*, *ip6_packet*, *addrconf*, *ip6_frag*... También se inician en este punto los protocolos de transporte *udpv6* y *tcpv6*.

5. PROGRAMACIÓN DE LOS CAMBIOS

Tras el montaje de la maqueta, los esfuerzos del proyecto se centraron en el estudio y comprensión del código de MIPv6 e IPv6, puesto que los dos están muy relacionados. Ésta fue una de las partes más costosas, ya que el código de MIPv6 es bastante extenso (únicamente el parche para el kernel ya son 19.700 líneas) y sus creadores no realizaron ningún documento explicativo.

Por motivos de simplicidad, en este proyecto se hizo que el HA fuera la entidad que realizara la funcionalidad de PA; de esta manera, el MN para registrar su estado activo y enviar las actualizaciones de localización lo hacía con el HA, ya era él el responsable de procesarlos (aunque en un submódulo diferente).

Para probar los cambios se trabajó sobre la maqueta que se había montado previamente, concretamente sobre el HA, el MN y el router. La única manera de debugar era mediante las funciones *printk*, propia del kernel, y *DEBUG*, propia del código MIPv6, lo que hacía que a veces fuera difícil averiguar de dónde provenían los errores. Además era frecuente que con determinados errores el sistema se colgara no dando tiempo a que los mensajes de debug aparecieran en el fichero */var/log/messages*. A esto se le ha de sumar lo comentado anteriormente de que cada vez que se modificaba algo del código de IPv6 y se compilaba, había que reiniciar el sistema para que los cambios surtieran efecto, puesto que el módulo de IPv6 no se puede descargar.

Básicamente la programación de los cambios consistió en implementar los mensajes de anuncio de área de paging, *Idle State Request* e *Idle State*

Reply.

5.1. Identificador de área de paging (PAI)

Para implementar este identificador, el draft de Paging IP proponía crear una nueva extensión a los paquetes ICMPv6 de *Router Advertisement* (RA). La ventaja de hacerlo de esta manera es que los RAs ya están creados y se usan para anunciar los prefijos de subred. Así pues, se modificó el código de envío de paquetes en el router añadiendo un nuevo campo que contenía el identificador de área de paging.

Una vez modificado el envío de RAs, se modificó también el procesado de éstos en recepción (código del nodo móvil) para que se pudiera entender la nueva extensión. El procesado de los RAs se realiza en el código de IPv6. El proceso seguido en recepción era: se obtiene el campo identificador de área de paging (PAI) y se compara con el valor anterior de PAI registrado. Si los valores eran iguales, esto indicaba que no se había cambiado de *Paging Area* y por tanto no se hacía nada; por el contrario, si eran diferentes se deducía que el nodo móvil había cambiado de *Paging Area*, y había que actualizar internamente el PAI y enviar al PAt (en nuestro caso, el HA) un mensaje *Location Update* (actualización de localización).

5.2. Location Update

Un nodo móvil debe actualizar su localización siempre que entra a una nueva área de paging. Esta actualización se realizaba mediante el envío de un mensaje *Idle State Request* (ISRQ), que contenía el PAI de la *Paging Area* en la que se encontraba el nodo móvil.

5.3. Idle State Request

El nodo móvil enviaba este mensaje a su *Paging Agent* cuando deseaba entrar en modo *idle* o cuando detectaba que había entrado en una nueva área de paging. Tanto los ISRQ como los *Idle State Reply* se implementaron como mensajes de movilidad *Mobility Headers*.

Los campos de opciones de los ISRQ eran: número de secuencia, PAI, idle, reserved. El número de secuencia se usaba para poder observar la concordancia entre peticiones *Idle State Request* y respuestas *Idle State Reply*. El campo PAI se enviaba en este paquete para informar al agente de paging del área de paging en la que el MN estaba en ese momento. Como este mismo mensaje (ISRQ) también es usado por el MN para entrar en estado durmiente, se añadió un campo al paquete llamado «*idle*» que diferenciaba estos dos casos conteniendo un valor diferente. Por último, se añadió un campo de reserva de bits para una futura ampliación del mensaje.



5.4.Location Update ACK

El PA cuando recibía las actualizaciones de localización del MN, debía confirmar su recepción. Esta confirmación se realizaba mediante el envío de un mensaje *Idle State Reply*.

5.5.Idle State Reply

Cuando el PA recibía un mensaje ISRQ debía confirmárselo al MN, mediante un mensaje *Idle State Reply* (ISRP). Como se ha comentado anteriormente, en el proyecto el HA realizó las funciones de PA. Estas funciones se emplazaron en un submódulo aparte.

El mensaje ISRP, implementado también como un mensaje de movilidad *Mobility Header*, contenía tres campos de opciones: número de secuencia, *status* y *reserved*. El campo *status* indicaba si la recepción del mensaje ISRQ era correcta. El número de secuencia era el mismo que el mensaje ISRQ al que hacía referencia, y el campo de *reserved* también era un campo de reserva de bits para un futuro.

En el código del nodo móvil también se hicieron modificaciones para poder procesar estos mensajes ISRP.

6. MAQUETA USADA EN EL PROYECTO

Los cambios programados especificados en el apartado anterior se probaron sobre la maqueta del proyecto.

Las fases de configuración de la maqueta pueden dividirse en 4. En un principio, en los ordenadores de la maqueta se instaló el sistema operativo Linux Red Hat 9. Posteriormente, se activó la funcionalidad IPv6 y se configuraron las funcionalidades MIPv6 de cada nodo (HA, MN y CN). Finalmente, en el router de la maqueta se instaló el *Router Advertisement Daemon* (radvd), necesario para anunciar los prefijos de red a los nodos conectados a la interfaz del router.

Los equipos usados en el montaje de la maqueta Mobile IPv6 son los siguientes: 3 PC de sobremesa (HA, Router y CN), 1 PC portátil (MN), 1 switch y 1 hub, y 5 tarjetas de red Ethernet. La distribución de los equipos con sus direcciones de red es la siguiente:

El prefijo de la *home network* es fec0:100:1000::/64. Pertenecen a la *home network* el HA y la interfaz eth0 del router. El prefijo de la *foreign network* es fec0:100:2000::/64. El CN y la interfaz eth1 del router pertenecen a la *foreign network*. El MN va moviéndose entre una red y otra.

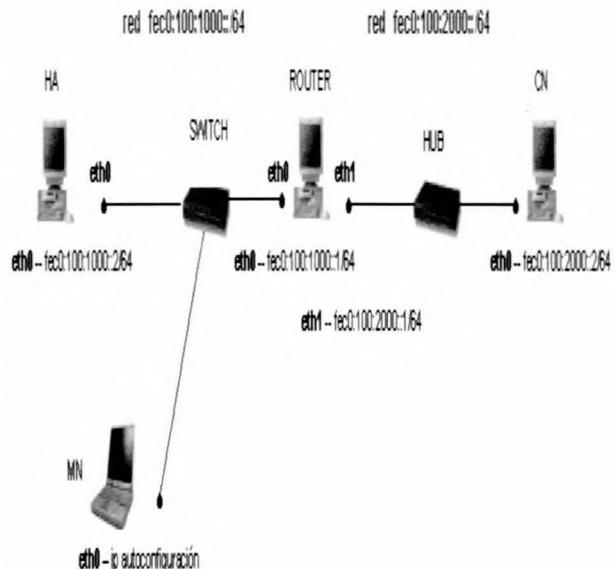


Figura 8. Distribución de los equipos de la maqueta

7.CONCLUSIONES

El presente artículo se ha explicado el protocolo MIPv6 y como puede a este se le puede añadir el soporte de nodos durmientes mediante paging IP. De una aproximación meramente teórica se ha pasado a una práctica centrada en estudio detallado de la implementación en código abierto de MIPv6 MIPL y de su interacción con IPv6 en el kernel de Linux. Esta parte ha supuesto el esfuerzo más importante del trabajo descrito y ha permitido realizar una primera modificación del código probada en una maqueta construida para tal efecto.

Se espera que en futuros trabajos esta modificación sea completada, por ejemplo separado el PA del HA, a la vez que se le añadan otras mejoras al código como la utilización de multicast para el envío de las *Paging Requests*. También se desea ampliar la maqueta introduciendo diferentes redes de acceso radio como por ejemplo 802.11 que permitan probar la efectividad del código desarrollado en una maqueta lo más parecida posible a una red 4G.

8.AGRADECIMIENTOS

Este trabajo ha sido financiado por el proyecto TIC2003-01748.

9. REFERENCIAS

- [1] M.V. de Diego, D. Gallego, J.A. López, A. Gómez. «UMTS: hacia una red todo IP». Comunicaciones de Telefónica I+D, nº 24. Enero 2002.
 - [2] 3rd Generation Partnership Project; «Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); IP Multimedia Subsystem (IMS); Stage 2". Diciembre 2003. 3GPP TS 23.228 version 5.11.0 Release 5.
 - [3] Deering, S. and R. Hinden. *Internet Protocol, Version 6 (IPv6) Specification*, RFC 2460, Diciembre 1998.
 - [4] Perkins, C. *IP Mobility Support for IPv4*, RFC 3220, Enero 2002.
 - [5] 3rd Generation Partnership Project 2; «Wireless IP Network Standard». Octubre 2002. 3GPP2 P.S0001-B v1.0.
 - [6] Dave Johnson, Charles Perkins, Jari Arkko. *Mobility Support in IPv6*, draft-ietf-mobileip-ipv6-24.txt, Julio 2003.
 - [7] Perkins, C. *IP Encapsulation within IP*, RFC 2003, Octubre 1996.
 - [8] Kempf, J., Editor. *Dormant Mode Host Alerting («IP Paging») Problem Statement*, RFC 3132, Junio 2001.
 - [9] Kempf, J., et. al. *Requirements and Functional Architecture for an IP Host Alerting Protocol*, RFC 3154, Agosto 2001.
 - [10] J. Kempf, Ed., «Dormant Mode Host Alerting (DMHA) Protocol Assessment», *Internet draft, draft-ietf-seamoby-paging-protocol-assessment-01.txt*, August 2002.
 - [11] Liebsch, M., Renker, G., and Schmitz, R.. *Paging Concept for IP based Networks*, draft-renker-paging-ipv6-01.txt, work in progress.
- Otra bibliografía consultada:
- IPv6 Working Group. Página web, URL <<http://www.ietf.org/html.charters/ipv6-charter.html>>
 - Seamless Mobility (Seamoby) Working Group y Mobile IP Working Group. Página web, URL <<http://www.ietf.org/html.charters/seamoby-charter.html>>
 - Mobile IPv6 Working Group. Página web, URL <<http://www.ietf.org/html.charters/mip6-charter.html>>
 - Linux IPv6 Router Advertisement Daemon (radvd). Página web, URL <<http://www.linuxhq.com/IPv6/radvd.html>>
 - Alessandro Rubini & Jonathan Corbet. *Linux device drivers 2nd edition*. Ed. O'Reilly.
 - *M IPL Mobile IPv6 for Linux*. Página web, URL <<http://www.mipl.mediapoli.com>>
 - The Linux Documentation Project. Página web, URL <<http://www.tldp.org/>>:
 - Henderson, Bryan. *Linux Loadable Kernel Module HOWTO*, <<http://tldp.org/HOWTO/Module-HOWTO/index.html>>
 - Rusling, David A. *The Linux Kernel* <<http://tldp.org/LDP/tlk/tlk.html>>
 - Bieringer, Peter. *Linux IPv6 HOWTO* <<http://tldp.org/HOWTO/Linux+IPv6-HOWTO/index.html>>
 - Jay, Peter and Pomerantz, Ori. *The Linux Kernel Module Programming Guide* <<http://tldp.org/LDP/lkmpg/lkmpg.pdf>>

8. AUTORES



Sara Berzosa, Ingeniera Técnica de Telecomunicaciones especialidad de telemática por la EPSC (UPC) desde el año 2003. Actualmente está cursando el primer curso del segundo ciclo de Ingeniería Superior de Telecomunicaciones en la misma EPSC.



Rafael Vidal, Ingeniero de Telecomunicaciones por la ETSETB (UPC) y profesor del Departamento de Ingeniería Telemática desde el año 2000, con docencia en la EPSC (UPC). Forma parte del grupo de investigación de redes inalámbricas desde el año 1998. Su ámbito de trabajo es el soporte a la movilidad en redes IP. Ha participado en diferentes proyectos de financiación pública y privada. Actualmente trabaja en los proyectos RUBI (Red Ubicua Basada en IP, TIC2003-01748) e I2Cat.