

# 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 Telecomunicaciones 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, sobretudo 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 que 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

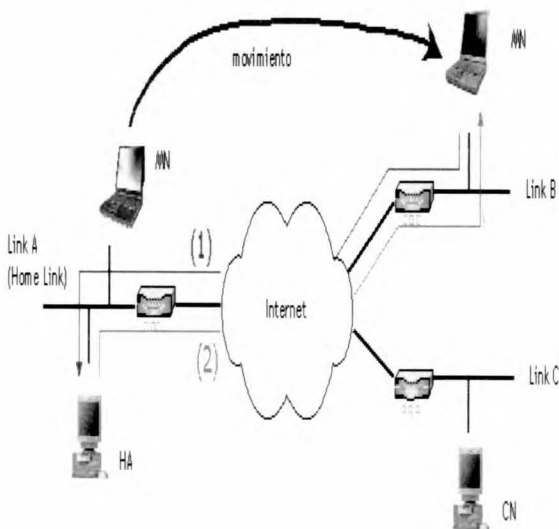


Figura 1. Proceso de registro con el HA

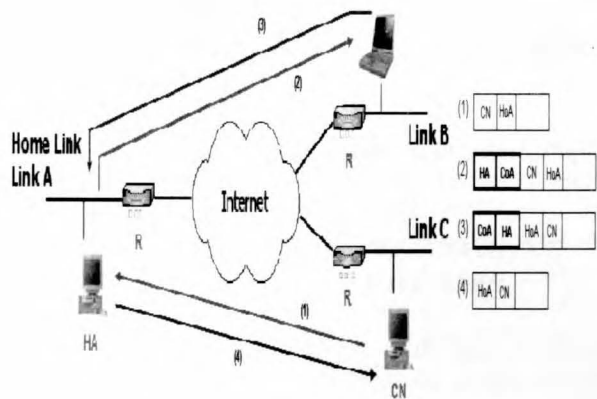


Figura 2. Tunneling bidireccional

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

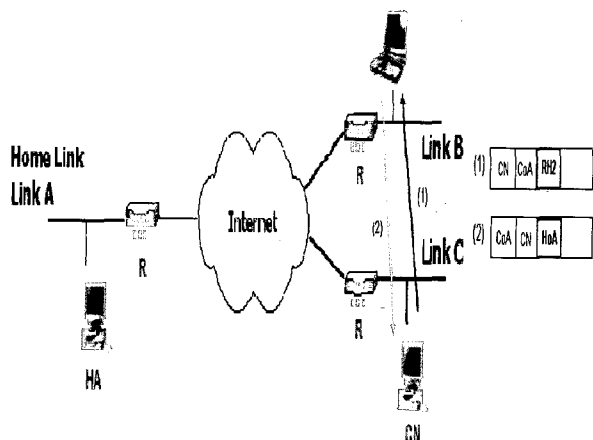


Figura 3. Route optimization

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 a cambia de área de paging y por tanto 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

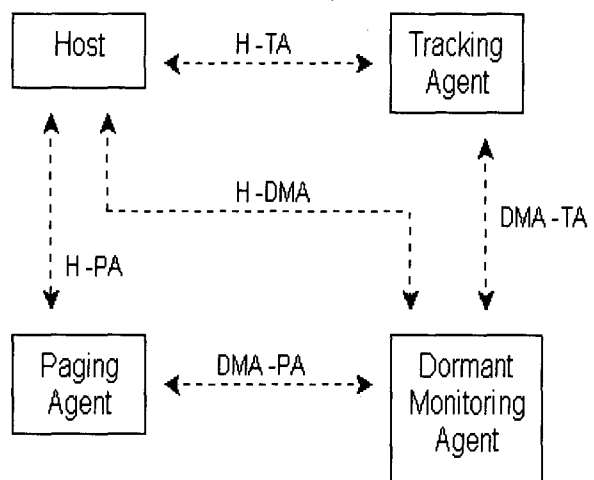


Figura 4. Arquitectura funcional paging IP

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

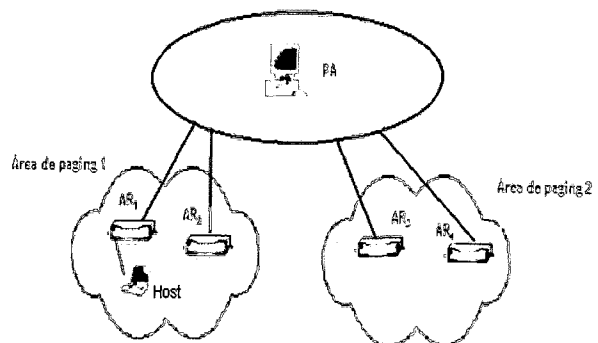


Figura 5. Escenario MIPv6 con paging IP

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.

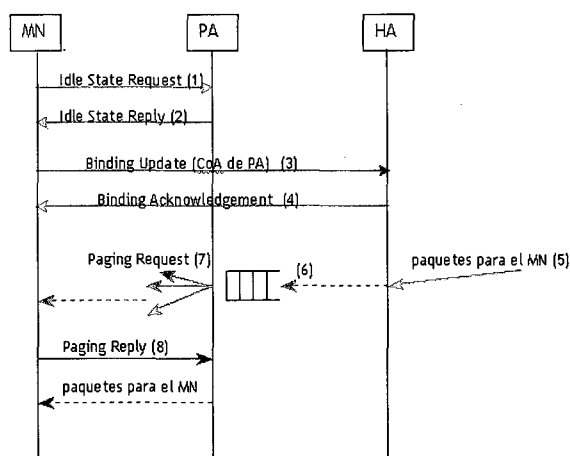


Figura 6. Proceso de registro y paging

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/syconfig/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 módulos 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>	71448	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

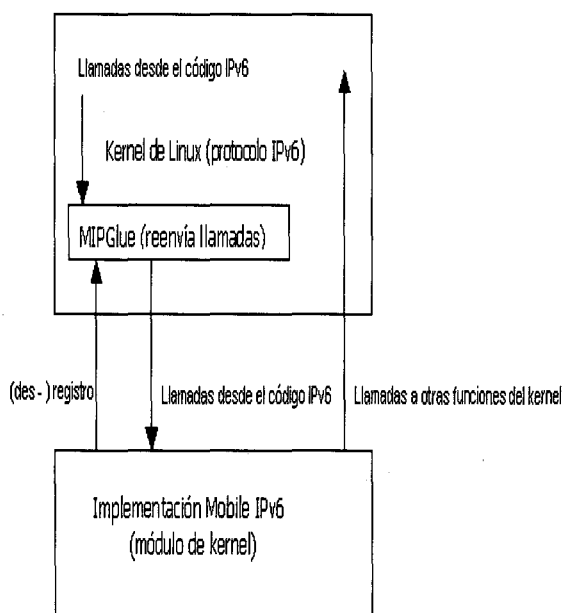


Figura 7. Esquema de comunicación entre IPv6, MIPv6 y el kernel

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* y *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 *udp6* y *tcp6*.

## 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 PA (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.

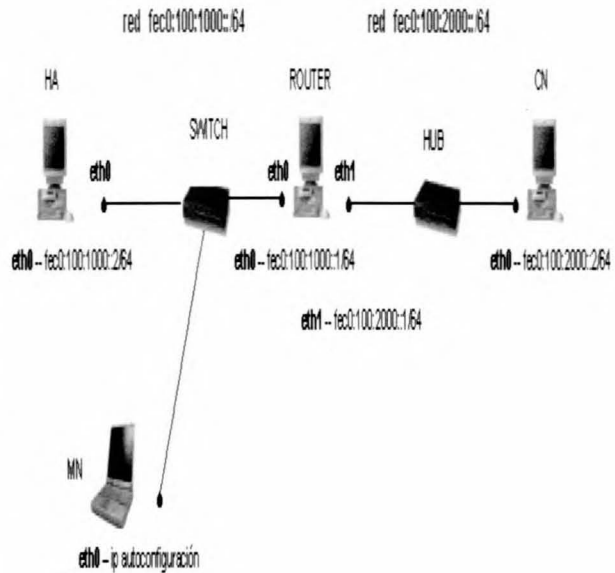


Figura 8. Distribución de los equipos de la maqueta

### 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

Esté 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, n° 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 2<sup>nd</sup> edition*. Ed. O'Reilly.
- MIPL 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.

