

# Montaje de la maqueta MIND sobre IPv6

Diego Chaparro dchaparro@acm.org  
Raúl Rodríguez rrodrigu@gsyc.escet.urjc.es

23 de Noviembre de 2001

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Montaje de la maqueta con IPv6</b>	<b>3</b>
2.1. Configuración del kernel con soporte para IPv6 y MobileIPv6 . . .	3
2.2. Configuración de las tarjetas PCMCIA 802.11b . . . . .	6
2.3. Diseño de la maqueta . . . . .	6
2.4. Configuración de las direcciones IPv6 . . . . .	7
2.5. Configuración de las rutas IPv6 . . . . .	8
2.6. Esquema detallado de la maqueta . . . . .	9
<b>3. Mobile IPv6</b>	<b>11</b>
3.1. Fundamentos de Mobile IPv6 . . . . .	11
3.2. Configuración de los nodos . . . . .	12
3.3. Puesta en funcionamiento . . . . .	13
<b>4. Cellular IPv6</b>	<b>15</b>
4.1. Fundamentos de Cellular IP . . . . .	15
4.2. Instalación de Cellular IPv6 . . . . .	16
4.3. Configuración de los nodos . . . . .	17
4.4. Puesta en funcionamiento . . . . .	18

## 1. Introducción

En este documento se explica el proceso de montaje de una maqueta de ordenadores, la cual servirá para probar protocolos de movilidad sobre IPv6, así como protocolos de micromovilidad en IPv6 como Cellular IPv6.

La maqueta tendrá un aspecto como el siguiente:

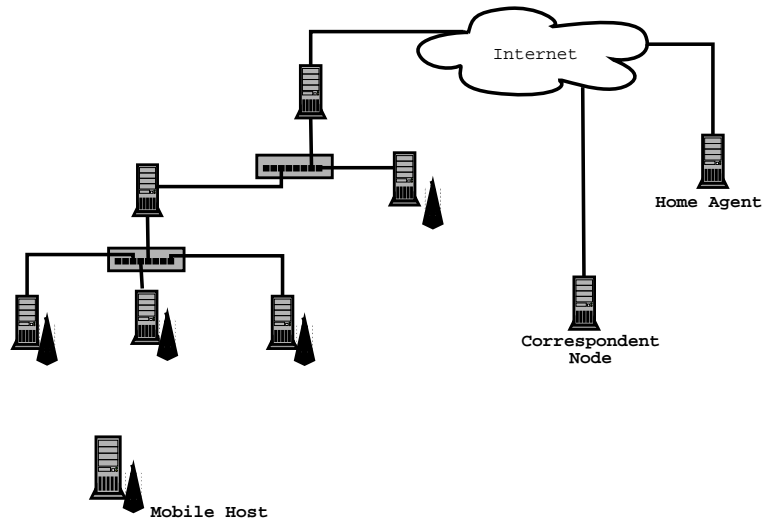


Figura 1

Básicamente la maqueta consiste en un conjunto de PCs a los que se conecta el Mobile Host dependiendo de su localización actual, y de este modo seguir teniendo conectividad independientemente del nodo al que se conecte y de manera transparente para el usuario ...

Para la comunicación entre el Mobile Host y los nodos a los que se conecta se utiliza tecnología wireless con tarjetas inalámbricas 802.11b.

Primero explicaremos el montaje de la maqueta para que funcione en IPv6, y después explicaremos la instalación de Mobile IPv6, Cellular IPv6, ... y los resultados obtenidos.

## 2. Montaje de la maqueta con IPv6

La maqueta consta de unos cuantos PCs de sobremesa (7 actualmente), y un portátil que actuará como Mobile Host. Todas estas máquinas tienen sistema operativo Debian GNU/Linux (woody) con kernel 2.4.7.

Algunas de estas máquinas (Mobile Host y nodos a los que se conecta) están provistas de tarjetas wireless 802.11b para darle mayor libertad de movimiento al Mobile Host. Todas estas tarjetas son PCMCIA, y en el caso de los PCs de sobremesa poseen un bridge PCI-PCMCIA para poder utilizarlas.

### 2.1. Configuración del kernel con soporte para IPv6 y MobileIPv6

A partir de la versión 2.2.19, el kernel de Linux incorpora soporte para IPv6. Con las series 2.4, se empiezan a incorporar nuevas funcionalidades hasta alcanzar el estado actual del módulo IPv6 del kernel, en estado experimental.

Entre las características que proporciona el módulo están:

- Espacio de direcciones ampliado
- Mecanismos de autenticación y privacidad
- Interoperabilidad con IPv4

El primer paso es bajarse los ficheros fuentes del kernel. Puedes usar la forma habitual de instalar paquetes en Linux (a partir de la herramienta `dselect` si tu distribución es Debian, o directamente a partir del fichero `.deb` también en Debian, o bien del `.rpm` si usas RedHat). También puedes conectarte a <http://kernel.org/> y bajártelos de allí. El lugar por defecto donde se instalarán los fuentes del kernel es `"/usr/src/kernel-source-2.x.x/"` (si estás en Debian).

A continuación hay que conectarse a la Helsinki University of Technology<sup>1</sup> y descargar la versión de Mobile IPv6 correspondiente con nuestro kernel (nosotros utilizaremos la versión del kernel 2.4.7): `mip6-0.9-v2.4.7.tar.gz`<sup>2</sup>

Un lugar para descomprimir el fichero puede ser `"/usr/local/"`, de la forma habitual:

```
host:/usr/local# tar xvfz mip6-0.9-v2.4.7.tar.gz
```

Esto creará el directorio `"/usr/local/mip6-0.9-v2.4.7"`. En él, se encuentra el fichero `mip6-0.9-v2.4.7.patch` que es el fichero con el parche para el kernel 2.4.7 (este parche funciona para los kernels 2.4.4 a 2.4.7) que introducirá las opciones para Mobile IPv6.

A continuación copia este fichero `“mip6-0.9-v2.4.7.patch”` al directorio donde están los fuentes del kernel que has instalado y ejecuta el parche con el siguiente comando:

---

<sup>1</sup><http://www.mipl.mediapoli.com/>

<sup>2</sup><http://www.mipl.mediapoli.com/download/mip6-0.9-v2.4.7.tar.gz>

```
host:/usr/local/mipv6-0.9-v2.4.7#: cd /usr/src/kernel-source-2.4.7
host:/usr/src/kernel-source-2.4.7#: patch -p1 < mipv6-0.9-v2.4.7.patch
```

Con esto ya están modificados los fuentes del kernel para soportar Mobile IPv6.

Lo siguiente que hay que hacer es arrancar una de las tres herramientas que te permiten recompilar el kernel. Para ello hay que situarse en el directorio anteriormente mencionado y ejecutar una de estas tres opciones:

- make menuconfig
- make oldconfig
- make xconfig

Nosotros utilizamos la tercera de las opciones, "make xconfig", que usa las X y es más vistosa.

En la sección **Networking Options** se encuentran las opciones relativas a IPv6 y Mobile IPv6. Las opciones que debes seleccionar (aquí aparecen las que nosotros tenemos seleccionadas para nuestras pruebas también con Mobile IPv4. En el fichero README de "/usr/local/mipv6-0.9-v2.4.7/" se especifican las que son estrictamente necesarias para Mobile IPv6) son las siguientes:

- Packet socket (Y)
- Kernel/User netlink socket (Y)
- Routing messages (Y)
- Networking packet filtering (replace ipchains) (Y)
- Socket filtering (Y)
- Unix domain sockets (Y)
- TCP/IP networking (Y)
- IP: multicasting (Y)
- IP: advanced router (Y)
- IP: policy routing (Y)
- IP: tunneling (Y)
- The IPv6 protocolo (EXPERIMENTAL) (m)
- IPv6: Mobility Support (EXPERIMENTAL) (m)
- MIPv6: Debug Messages (m)

Es conveniente poner las opciones de Mobile IPv6 como módulos en vez de incluirlos en el kernel directamente para prevenir posibles fallos a la hora de ejecutar el kernel las primeras veces.

La opción `MIPv6: AH Support` proporciona soporte para *IPSec Authentication Header* (mecanismos de seguridad en Mobile IPv6). Si quieres usar esta opción selecciónala (los nodos compilados con esta opción no pueden interactuar con los nodos que no la hayan seleccionado).

Con las opciones ya marcadas, sólo queda guardar los cambios y recompilar el kernel:

- `make dep`
- `make clean`
- `make bzImage`
- `make modules`
- `make modules_install`

Ya está preparado el kernel para poder funcionar con IPv6 y MobileIPv6. Sólo queda actualizar LILO. Para ello hay que hacer una copia de la imagen del kernel recién compilado (bajo `"/usr/src/kernel-source-2.4.7/arch/i386/boot/bzImage"`) dentro del directorio con las imágenes de kernel habitual (normalmente en `"/boot"`):

```
host:/usr/src/kernel-soruce-2.4.7/#: cp arch/i386/boot/bzImage /boot/vmlinuz-2.4.7
```

Ahora hay que añadir una nueva entrada en el fichero `"/etc/lilo.conf"` con el siguiente texto:

```
...
image=/boot/vmlinuz-2.4.7
label=Linux-2.4.7
readonly
...
```

Si se han completado todos estos pasos satisfactoriamente, lo único que queda es ejecutar `lilo` desde un terminal (para que los cambios realizados en `"/etc/lilo.conf"` tengan efecto) y reiniciar la máquina. Cuando aparezca *LILO* escribiremos `Linux-2.4.7` y ya tendremos nuestro equipo configurado con IPv6 y Mobile IPv6.

Antes de reiniciar la máquina, hay que ejecutar el siguiente comando para completar la configuración de Mobile IPv6:

```
host:~/: mknod /dev/mipv6_dev c 0xf9 0
```

## 2.2. Configuración de las tarjetas PCMCIA 802.11b

Las tarjetas de las que disponemos son varias Lucent Technologies, y algunas Compaq pero que también tienen el chip de Lucent.

Hay varios modos para configurar las tarjetas wireless, se puede configurar para que puedan comunicarse con un Access Point o se pueden configurar en modo Ad-Hoc, para que puedan comunicarse las tarjetas entre ellas. Este último modo es el que nos interesa a nosotros para que pueda hablar el Mobile Host con los nodos a los que se conecta.

Para hacer funcionar la tarjeta wireless necesitamos instalar el paquete PCMCIA, que lo podemos conseguir aquí<sup>3</sup>, y lo instalamos de la manera tradicional (make config, make all, make install).

Con este paquete vienen varios drivers que se pueden utilizar para estar tarjetas, wlan y orinoco, pero nosotros utilizamos el orinoco que es el más reciente.

También debemos instalar el paquete wireless tools<sup>4</sup>, que nos proporcionará herramientas para la configuración de las tarjetas.

Para configurar las tarjetas en modo Ad-Hoc necesitamos editar el fichero /etc/pcmcia/wireless.opts, en ese fichero hay varios apartados para configurar la tarjeta dependiendo del modelo (en realidad depende de la dirección MAC), y el que nos interesa a nosotros es el siguiente:

```
# Lucent Wavelan IEEE
# Note : wlan_cs driver only, and version 1.0.4+ for encryption support
# Nota: Anado la última dirección MAC para las Compaq
*,*,*,00:60:1D:*|*,*,*,00:02:2D:*|*,*,*,00:02:A5:*)
  INFO="Wavelan IEEE example (Lucent default settings)"
  ESSID="MINIRED"
  MODE="Ad-Hoc"
  RATE="auto"
```

En realidad lo que hace es mirar la dirección MAC de las tarjetas y dependiendo de los 24 primeros bits entra en una de las secciones. Nosotros hemos añadido la última de las direcciones MAC para que tengamos la misma configuración en las tarjetas Lucent (las dos primeras direcciones MAC) que en las Compaq.

Con esto ya tenemos configuradas en modo Ad-Hoc para que puedan hablar entre ellas.

## 2.3. Diseño de la maqueta

Para la construcción de la maqueta necesitamos definir las subredes que queremos crear, los prefijos de red que vamos a usar, las direcciones de cada subred ...

Con la maqueta que hemos diseñado tenemos 6 subredes distintas, lo que nos permitirá posteriormente bastantes posibilidades para desarrollar nuestras pruebas:

---

<sup>3</sup><http://sourceforge.net/projects/pcmcia-cs/>

<sup>4</sup>[http://www.hpl.hp.com/personal/Jean\\_Tourrilhes/Linux/Tools.html](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html)

**Figura 2**

Y les asignamos las siguientes direcciones de red a cada una de ellas:

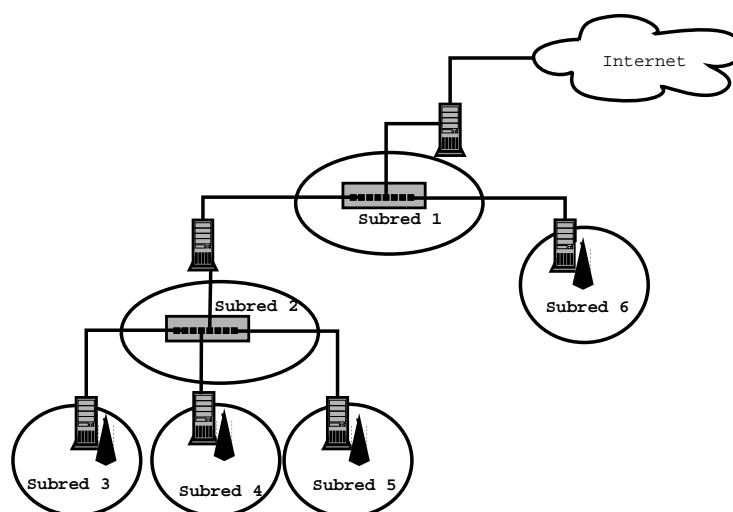
- **Subred 1:** fec0:0:0:2::/64
- **Subred 2:** fec0:0:0:3::/64
- **Subred 3:** fec0:0:0:4::/64
- **Subred 4:** fec0:0:0:5::/64
- **Subred 5:** fec0:0:0:7::/64
- **Subred 6:** fec0:0:0:6::/64

#### 2.4. Configuración de las direcciones IPv6

Cada una de las máquinas de la maqueta (excepto el Mobile Host) tiene varias interfaces de red, uno de ellos le conecta con la parte superior del árbol y otra interfaz con las máquinas que hay por debajo. Gracias a esta estructura hemos definido las direcciones de la siguiente forma:

- **Interfaz superior** (la que le conecta al nodo padre): Adquiere la dirección IPv6 que le asigna el padre mediante radvd.
- **Interfaz inferior:** Tiene una dirección IPv6 fija, y tiene funcionando radvd para asignar direcciones a las máquinas que se conecten a esa subred.

El radvd ( Router advertisement ) es un demonio que informa a los nodos que se conectan a la subred cuál es la dirección de esa subred, y cuál es su prefijo. Un ejemplo de un mensaje que manda el radvd es el siguiente:



```

Router advertisement from fe80::250:4ff:fe47:d29a (hoplimit 255)
  AdvCurHopLimit: 64
  AdvManagedFlag: off
  AdvOtherConfigFlag: off
  AdvHomeAgentFlag: off
  AdvReachableTime: 0
  AdvRetransTimer: 0
  Prefix fec0:0:0:1::/64
    AdvValidLifetime: 2592000
    AdvPreferredLifetime: 604800
    AdvOnLink: on
    AdvAutonomous: on
    AdvRouterAddr: off
  AdvSourceLLAddress: 00 50 04 47 D2 9A

```

Este mensaje nos indica que lo está mandando una máquina con dirección fe80::250:4ff:fe47:d29a y que está avisando de que la subred tiene una dirección de tipo: fec0:0:0:1::/64. Y el fichero de configuración del radvd en /etc/radvd.conf para que mande ese tipo de mensajes sería el siguiente:

```

interface eth0
{
    AdvSendAdvert on;
    MaxRtrAdvInterval 10;
    #AdvSourceLLAddress off;
    prefix fec0:0:0:1::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
    };
};

```

Cuando una máquina que se conecta a esa subred recibe un mensaje de ese tipo se configura automáticamente una dirección adecuada para esa subred. Y para ello lo que hace es coger la parte de dirección de red que recibe y después le adjunta la dirección MAC de su tarjeta. Por ejemplo una máquina con dirección MAC 00:50:DA:4F:A7:87 obtiene la siguiente dirección en esa subred: fec0::1:250:daff:fe4f:a787.

La configuración de las direcciones de red que queremos poner fijas se indican en el fichero /etc/network/interfaces, y la estructura que tienen las máquinas de la maqueta es la siguiente (ejemplo en la subred 4):

```

iface eth2 inet6 static
    address fec0::5:202:a5ff:fe6e:5209
    netmask 64

```

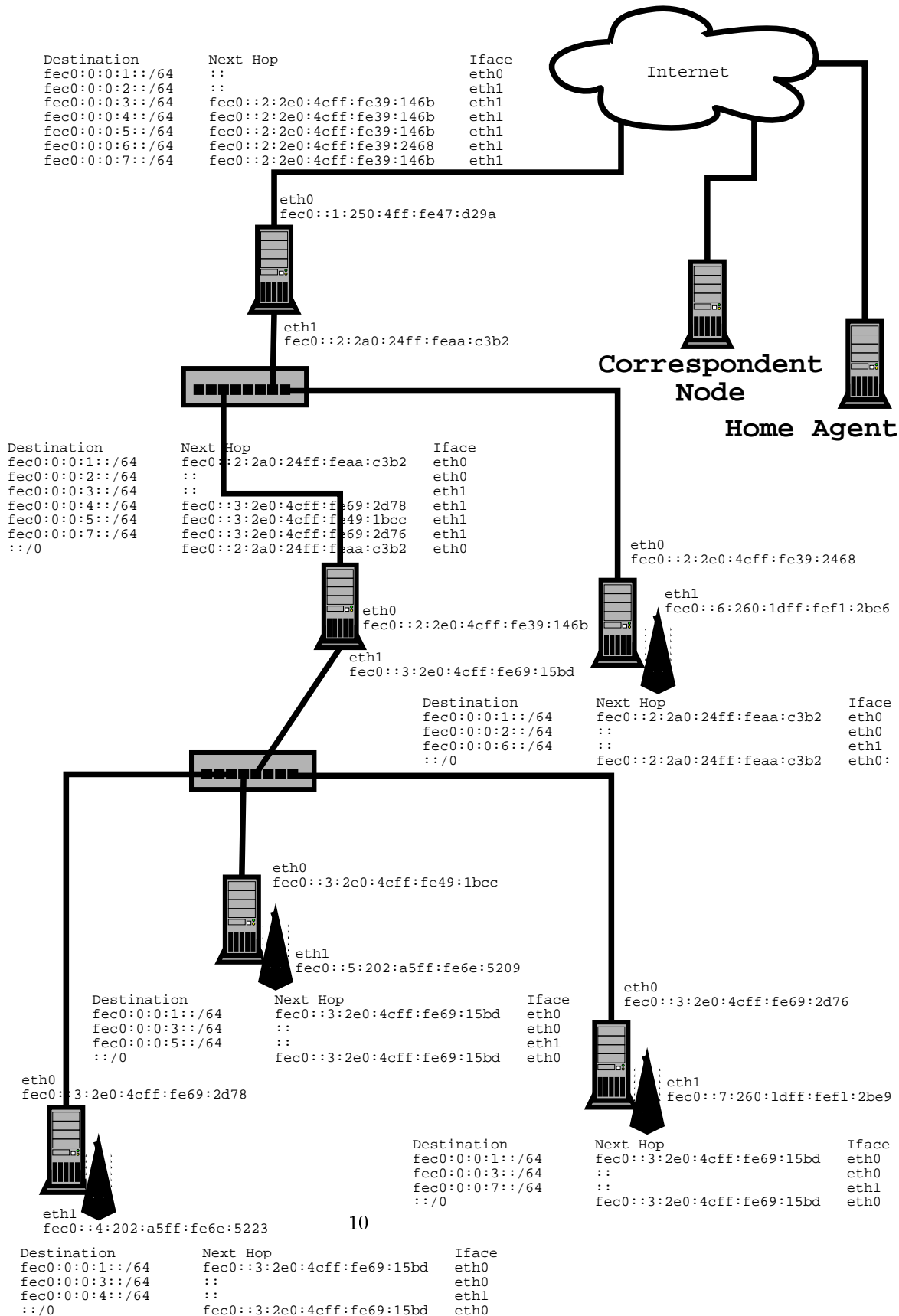
## 2.5. Configuración de las rutas IPv6

Para la configuración de las rutas hemos utilizado un script que se ejecuta en el arranque en el que le indicamos todas las rutas IPv6 que tiene cada máquina. La forma de indicar las rutas en IPv6 es similar a como se hace en IPv4. Un ejemplo:

```
route add -A inet6 fec0:0:0:3::/64 gw fec0::2:2e0:4cf:fe39:146b dev eth1
```

## **2.6. Esquema detallado de la maqueta**

A continuación mostramos con detalle el esquema de la maqueta, mostrando las direcciones IPv6 de cada máquina y las rutas más significativas que tienen:



## 3. Mobile IPv6

### 3.1. Fundamentos de Mobile IPv6

Mobile IP especifica las operaciones de los dispositivos móviles usando IPv6. Cada dispositivo móvil es siempre identificado por su dirección “Home Address”, independientemente de su actual situación en Internet. Cuando el dispositivo móvil se encuentra alejado de su “Home Network”, éste se asocia a una nueva “care-of address”, la cual nos da información sobre la actual situación del dispositivo móvil. Los paquetes IPv6 dirigidos a la “Home Address” del dispositivo móvil son transparentemente encaminados hasta su “Care-of address”.

Diferencias más destacables entre Mobile IPv4 y Mobile IPv6:

- Lo que se conocía en Mobile IPv4 como “Route optimization” ahora forma parte del protocolo en Mobile IPv6. Esto significa que los paquetes dirigidos desde el “Correspondent Node” hasta el “Mobile Node” lo hacen directamente y no tienen que pasar por el “Home Agent” y luego ser reenviados hasta el “Mobile Node”.
- En este protocolo los paquetes que manda el “Mobile Node” llevan como dirección origen la “Care-of address” en la cabecera IP, y luego llevan una opción para el destino con la dirección “Home Address”. Esto, a diferencia de Mobile IPv4, hace que sea transparente para todos los encaminadores y para las capas superiores.
- El uso de la “Care-of address” como la dirección origen de los paquetes IP simplifica el encaminamiento de paquetes multicast enviados por el “Mobile Node”. En Mobile IPv4, el “Mobile Node” tenía que hacer un túnel hasta su “Home Agent” para poder usar de forma transparente su “Home Address” como dirección origen de los paquetes multicast. En Mobile IPv6, con la opción de destino de la “Home Address” permite ser compatible con el encaminamiento multicast, que en parte está basado en el dirección origen del paquete.
- Ya no hace falta tener encaminadores especiales que hagan de “Foreign Agent” como en Mobile IPv4. Ahora, el Mobile Node usa las características que le proporciona IPv6, tales como autoconfiguración de dirección y “Neighbor Discovery”.
- La mayoría de los paquetes que se envían a un Mobile cuando no está en la “Home Network” se hace usando una cabecera de encaminamiento IPv6 (IPv6 Routing Header) en lugar de encapsulación de IP (como en Mobile IPv4), y esto hace que se reduzcan los bytes necesarios en la cabecera.
- Cuando un Mobile Node no está en la “Home Network”, el “Home Agent” intercepta cualquier paquete que se dirige hacia el “Mobile Node” usando “IPv6 Neighbor Discovery” en lugar de ARP como en Mobile IPv4. Esto simplifica la implementación de Mobile IP al ser independiente de la capa de enlace, cosa que no ocurre con ARP.

- El mecanismo de descubrimiento de la dirección del “Home Agent” dinámica en Mobile IPv6 usa IPv6 anycast y devuelve una sola respuesta al “Mobile Node”, al contrario de Mobile IPv4 que usaba mensajes de broadcast y una respuesta de cada uno de los “Home Agents”. El mecanismo de Mobile IPv6 es más eficiente y más seguro.

## 3.2. Configuración de los nodos

Para el uso de Mobile IPv6 necesitamos tener los siguientes elementos:

- **Un Mobile Node:** Este es el dispositivo que se va a “mover” entre distintas subredes. Siempre tiene una “Home Network” en la que se encuentra su “Home Agent”, y tiene una dirección de esa subred que va a ser la dirección por la cual va a ser accesible en todo momento independientemente de localización (“Home address”).
- **Home Agent:** Es el encargado de recoger los paquetes que son enviados al “Mobile Node” y reenviárselos a su actual “Care-of address” cuando éste no se encuentra en la “Home Network”.
- **Correspondent Node:** Es un nodo que se encuentra en cualquier lugar de Internet, y es el nodo con el cual se está comunicando el “Mobile Node”.

En cada uno de estos nodos debemos tener instalados correctamente el módulo de Mobile IPv6, tal como se indica en la sección 2.1. Para la configuración de dicho módulo existen varios ficheros de configuración:

- */etc/sysconfig/network-mip6.conf:* Fichero principal de configuración
- */etc/mipv6\_acl.conf:* Lista de control de acceso de Mobile Nodes.
- */etc/mipv6\_sas.conf:* Seguridad de Mobile IPv6.

La configuración de cada uno de los nodos sería la siguiente:

- **Mobile Node.** El fichero */etc/sysconfig/network-mip6.conf* contiene lo siguiente:

```
# MIPL Mobile IPv6 Configuration file

FUNCTIONALITY=mn
DEBUGLEVEL=7
# TUNNEL_SITELOCAL=yes
HOMEADDRESS=fec0::1:260:1dff:fe1:2be8/64
HOMEAGENT=fec0::1:250:4ff:fe47:d29a
# MOBILENODEFILE=/etc/mipv6_acl.conf
RTR_SOLICITATION_INTERVAL=1
RTR_SOLICITATION_MAX_SENDTIME=5
```

- **Home Agent.** El fichero */etc/sysconfig/network-mip6.conf* contiene lo siguiente:

```
# MIPL Mobile IPv6 Configuration file

FUNCTIONALITY=ha
DEBUGLEVEL=1
TUNNEL_SITELOCAL=yes
# HOMEADDRESS=fec0::1:260:1dff:fe47:2be8
# HOMEAGENT=fec0::1:250:4ff:fe47:d29a
MOBILENODEFILE=/etc/mipv6_acl.conf
# RTR_SOLICITATION_INTERVAL=1
# RTR_SOLICITATION_MAX_SENDDTIME=5
```

Y el fichero `/etc/sysconfig/mipv6_acl.conf`:

```
ALLOW fec0::1:260:1dff:fe47:2be8/64
```

- **Correspondent Node.** El fichero `/etc/sysconfig/network-mip6.conf` contiene:

```
# MIPL Mobile IPv6 Configuration file

FUNCTIONALITY=cn
DEBUGLEVEL=2
# TUNNEL_SITELOCAL=yes
# Home address for mobile node with prefix length. Example:
# HOMEADDRESS=3ffe:b00:c18:1fff:0:0:0:bd5
# HOMEAGENT=3ffe:b00:c18:1fff:0:0:0:3cb
# MOBILENODEFILE=/etc/mipv6_acl.conf
# MD5KEY=
# SHA1KEY=
# RTR_SOLICITATION_INTERVAL=1
# RTR_SOLICITATION_MAX_SENDDTIME=5
```

Aparte de estos elementos necesitamos que los encaminadores de las subredes a las que se conecte el “Mobile Node” dispongan de un `radvd` ejecutando, como se explicó en la sección 2.4.

### 3.3. Puesta en funcionamiento

Una vez configurados todos los nodos, solamente debemos activar los módulos de Mobile IPv6. Para ello en cada una de las máquinas (Mobile Node, Home Agent y Correspondent Node) debemos hacer lo siguiente:

```
/etc/init.d/mobile-ip6 start
```

Y con esto ya estarán los nodos preparados para empezar a funcionar.

Si el “Mobile Node” se encuentra en la “Home Network” el funcionamiento de éste será el normal, podrá enviar y recibir paquetes como si fuera un nodo convencional.

Cuando el “Mobile Node” no se encuentra en la “Home Network” lo primero que hace es adquirir una nueva dirección de la subred a la que se conecta (“Care-of address”). Después de esto necesita que su “Home Agent” conozca

esa nueva dirección, y para ello le manda un “Binding Registration” (paquete con opción para el destino “Binding Update”), y el Home Agent le responde con una paquete con opción para el destino “Binding Acknowledgement”. A partir de ese momento esa nueva dirección será la “primary care-of address”, y el “Home Agent” interceptará los paquetes dirigidos hacia el “Mobile Node” mediante proxy Neighbor Discovery, y se los enviará mediante “IPv6 encapsulation”. Cada vez que el “Mobile Node” se cambie de subred mandará al “Home Agent” un “Binding Update”.

Podemos probar a conectar el “Mobile Node” a cualquiera de las subredes que tenemos y veremos que va adquiriendo nuevas direcciones y puede seguir comunicándose con el “Correspondent Node” mediante algún programa que tenga soporte para IPv6 (ssh, ping6, ...). En la siguiente dirección<sup>5</sup> podemos ver los programas que tienen soporte para ipv6 en debian.

---

<sup>5</sup><http://people.debian.org/~csmall/ipv6/packages.html>

## 4. Cellular IPv6

### 4.1. Fundamentos de Cellular IP

Cellular IP es un protocolo de micro-movilidad que se centra en la movilidad de los nodos en pequeñas zonas que se encuentran situadas estratégicamente dentro de una determinada jerarquía de nodos, a la vez que interactúa con Mobile IP para soportar el movimiento de nodos entre grandes áreas de la red. Cellular IP es una extensión al protocolo de Mobile IP, y no una alternativa a este.

Es decir, podemos tener una estructura jerárquica de nodos como la siguiente:

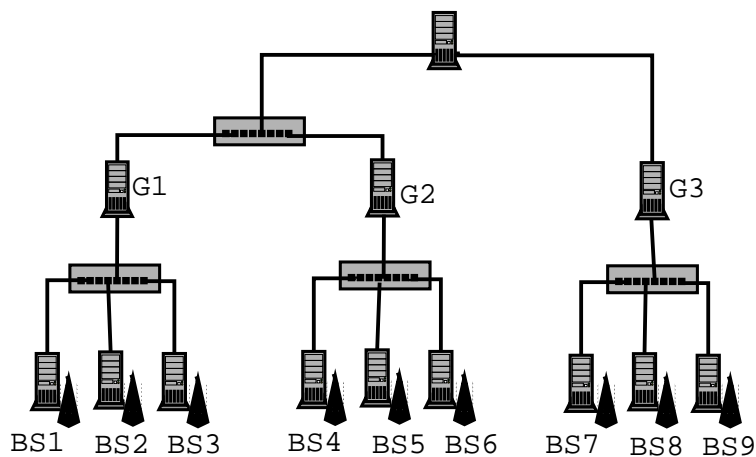


Figura 3

En esta estructura de red, Cellular IP se centra en la movilidad de nodos móviles entre las partes bajas de la jerarquía que comparten un mismo nodo padre (diferentes Base Station), mientras que Mobile IP soportaría cualquier otro movimiento de los nodos móviles (de una subred Cellular IP a otra).

En Cellular IP intervienen los siguientes actores:

- Gateway: Forma parte de la jerarquía de la subred Cellular IP, siendo el nodo padre de esta jerarquía. Es el encargado de dirigir los paquetes con destino el MH (Mobile Host) hacia la BS (Base Station) adecuada dependiendo de la posición actual del MH y viceversa, encaminar los paquetes procedentes del MH hacia Internet..
- Base Station: Es el encargado de comunicarse con el MH. Encamina los paquetes procedentes del MH hacia Internet y los paquetes dirigidos al MH procedenes de Internet.
- Mobile Host: Es el nodo móvil que se mueve entre las diferentes subredes, y se conecta en cada momento con la Base Station adecuada.

Cuando el MH se encuentra en una subred Cellular IP, éste utiliza como dirección origen de sus paquetes una dirección de la subred del GW, que ha sido autoconfigurada a través de unos mensajes (beacons) que envían las BS. Esta dirección es la que se utilizará como “Care-of address” de Mobile IP. El encaminamiento dentro de la subred Cellular IP se realiza dependiendo de la situación actual del MH, y esta información se actualiza cada vez que el MH manda algún paquete, y esta información es conocida por el Gateway para que pueda encaminar los paquetes correctamente. Cuando el MH cambia de BS esta información es reconocida por el GW cuando los paquetes de datos del MH pasan a través de él, y así podrá encaminar correctamente los paquetes con destino el MH.

Por ejemplo, en la maqueta anterior si el MH está conectado a la BS2, los paquetes de datos que envíe el MH pasarán por la BS2, después por el G1, y después se encaminarán hacia Internet. Estos paquetes llevarán una opción de cabecera para el destino de Mobile IP para indicar cual es la Home Address de este nodo, para que el Correspondent Node sepa cual es la dirección real del nodo, que no corresponde con la dirección origen del paquete, que es la que adquiere en la subred Cellular IP. En cuanto a Mobile IP, éste se comporta como si el MH estuviera en la subred del G1 y se encarga de encaminar los paquetes precisamente hasta esa subred, después ya se encarga Cellular IP de encaminar los paquetes hacia la posición actual del MH dentro de la subred Cellular IP.

## 4.2. Instalación de Cellular IPv6

La instalación de Cellular IPv6 es más sencilla que la de Mobile IPv6. Los requerimientos del sistema son los siguientes:

- Kernel a partir de 2.2.12. Nosotros hemos utilizado el 2.4.7.
- Tener instalado el paquete iproute.
- Tener instaladas las librerías libpcap con soporte ipv6.
- Se pueden usar tarjetas wireless que soporten funciones SPY, para medir la calidad de la señal.

Podemos bajarnos la implementación de Cellular IPv6 de la siguiente dirección: [http://w3dpdext1.intranet.gr/cipv6/download\\_area.htm](http://w3dpdext1.intranet.gr/cipv6/download_area.htm)

La instalación es muy sencilla. Primero descomprimos los fuentes: `tar xvzf cipv6-1.1.tar.gz`

Una vez descomprimidos los fuentes vemos que se crean dos directorios: `cipmobile6` y `cipnode6`. En el primero de ellos está el código que se ejecutará en el Mobile Host y en el segundo está el código que se ejecutará en todos los demás nodos, Base Stations y Gateway. Después entramos en el directorio correspondiente, ejecutamos: `make` y ya se compilarán los fuentes necesarios.

### 4.3. Configuración de los nodos

La configuración de los nodos es la siguiente:

- En el Mobile Host:

La configuración del Mobile Host se especifica en el fichero cipmobile6.conf, en este fichero se especifica la interfaz que usa el Mobile Host y algunos parámetros de tiempo. En nuestro caso:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
wireless interface=          eth0
air interface name=         wavelan
route-update-time=          3000      %in milliseconds
paging-update-time=         30000     %in milliseconds
active-state-timeout=       9000      %in milliseconds
handoff=                     1        %forced (=0) or SNR based (=1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

- En las Base Station:

La configuración de las Base Station se especifica en el fichero cipnode6.conf. La configuración de las Base Station de la maqueta es la siguiente:

- Base Station 1

```
GW:                            NO
IF YES, default router's IP address: \%(wire, eth0, 3ffe:2d00:24:100::1)
IF NO, neighbor, uplink direction:  (wire, eth0, fec0::3:2e0:4cff:fe69:15bd)
leaf neighbours(s):             (wireless, eth2)
paging cache:                    YES
route-timeout:                   1000   \%in milliseconds
paging-timeout:                  60000  \%in milliseconds
max number of mobiles in cache:  100
max number of node interfaces:  10
Base Station ID:                 1
Paging Area ID:                  1
CIP Network ID:                  1
```

- Base Station 2

```
GW:                            NO
IF YES, default router's IP address: \%(wire, eth0, fec0::3:2e0:4cff:fe69:15bd)
IF NO, neighbor, uplink direction:  (wire, eth0, fec0::3:2e0:4cff:fe69:15bd)
leaf neighbours(s):             (wireless, eth1)
paging cache:                    YES
route-timeout:                   1000   \%in milliseconds
paging-timeout:                  60000  \%in milliseconds
max number of mobiles in cache:  100
max number of node interfaces:  10
Base Station ID:                 2
Paging Area ID:                  1
CIP Network ID:                  1
```

- En el Gateway: La configuración del Gateway se especifica en el fichero `cipnode.conf`:

```

GW:                                YES
IF YES, default router's IP address: (wire, eth0, fec0::2:2a0:24ff:feaa:c3b2)
IF NO, neighbor, uplink direction:
leaf neighbours(s):                (wire, eth1, fec0::3:2e0:4cff:fe69:2d78)
(wire, eth1, fec0::3:2e0:4cff:fe49:1bcc)
paging cache:                      YES
route-timeout:                      1000  %in milliseconds
paging-timeout:                     60000 %in milliseconds
max number of mobiles in cache:     100
max number of node interfaces:      10
Base Station ID:
Paging Area ID:                     1
CIP Network ID:                     1

```

#### 4.4. Puesta en funcionamiento

Como hemos dicho anteriormente, Cellular IP es un protocolo de micromovilidad, por lo tanto debería ir acompañado de un protocolo de macromovilidad como Mobile IP. Por tanto en nuestra maqueta se usa Mobile IPv6 junto con Cellular IPv6, pero para su correcto funcionamiento deberían interoperar perfectamente y esto no es así. Para que interoperen entre los dos hemos tenenos que hacer algunas operaciones extra al ponerlos en funcionamiento. Una de las razones por las que no interoperan correctamente es la siguiente: El Gateway de Cellular IP emite unos paquetes (Gateway Broadcast Packet) que permiten a las Base Station conocer el prefijo de subred del Gateway y utilizar esta información a la hora de enviar los beacons que reciben los Mobile Hosts y que les permite autconfigurar sus direcciones. Esto lo hace correctamente. Pero a la vez que ocurre ésto, MobileIP debería darse cuenta de esa nueva dirección y hacer que esa dirección fuera su Care-Of-Address, pero esto no ocurre así. Para arreglar esto lo que hacemos es poner en las Base Station un `radvd` para que emitan el prefijo de subred del Gateway.

```

interface eth2
{
    AdvSendAdvert on;
    MaxRtrAdvInterval 10;
    #AdvSourceLLAddress off;
    prefix fec0:0:0:3::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
    };
};

```

Con esto ya conseguimos que el módulo de Mobile IP en el Mobile Host se entere de esa nueva dirección, la use como su Care-of-address y se registre en el Home Agent correctamente.

Los pasos a seguir para poner la maqueta en funcionamiento son los siguientes (si no lo hacemos en este orden a veces ocurren errores extraños):

- Arrancar la maqueta de Cellular IP sin radvds en las Base Station.
- Arrancar Mobile IP en el MN y llevarlo a la Home Network para que adquiriera la dirección Home Address.
- Llevarlo de nuevo a la Foreign Network.
- Arrancar los radvds en las Base Station.
- Después de esto el MN debe registrarse correctamente en el HA, y ya podemos hacer handoffs entre las Base Station.