Universidad Carlos III de Madrid

Escuela Politécnica Superior

Grado en Ingeniería Telemática

Estudio de soluciones de movilidad a nivel de enlace y nivel de red

Trabajo Fin de Grado

Autor:	**Miriam Marciel Noguera**
Tutor:	**Carlos Jesús Bernardos Cano**
Director:	**Pablo Serrano Yáñez-Mingot**

Septiembre de 2012

Trabajo Fin de Grado

Estudio de soluciones de movilidad a nivel de enlace y nivel de red

Autor

Miriam Marciel Noguera

Tutor

Dr. Carlos Jesús Bernardos Cano

Director

Dr. Pablo Serrano Yáñez-Mingot

Realizado el acto de defensa y lectura del Trabajo Fin de Grado el día de Septiembre de 2012 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, el tribunal:

PRESIDENTE:

SECRETARIO:

VOCAL:

acuerdan otorgarle la calificación de:

CALIFICACIÓN:

Leganés, a de Septiembre de 2012

A mis padres, a mi hermana y a mis abuelos

Agradecimientos

Lo primero dar las gracias a mis padres y a mi hermana, que han los que siempre me han apoyado y me han hecho ser la persona que soy. También dar las gracias a mis abuelos que siempre me han dado todo su cariño y apoyo.

Agradecer a mi tutores, Carlos y Pablo, por todo vuestro tiempo, por toda vuestra ayuda y por enseñarme el camino de la investigación.

Dar las gracias a toda la gente he ido conociendo en la universidad a lo largo de estos años, especialmente a Sara, Ricardo y Javi. Hemos disfrutado de muchos momentos, algunos han sido un poco más duros, pero siempre conseguíamos encontrar la manera de hacerlo más ameno.

Tampoco olvidarme de Dani. Sé que en ocasiones te he tenido un poco abandonado pero ya sabes que me faltaba tiempo. Gracias por tu apoyo durante estos años!

Y como no, agradecer a ese grupo de amigos que es difícil de mejorar: Alber, Ana, Carlos, Diana, Helena, María, Mati, Raquel, Toni y Vicky. Gracias por todos los momentos que hemos vivido todos estos años y por escucharme, aunque muchas veces no me entendieseis. También dar la bienvenida al "peque" que está en camino y que nos va a hacer a todos tíos postizos.

La vida no es fácil, para ninguno de nosotros. Pero... ¡qué importa!
Hay que perseverar y, sobre todo, tener confianza en uno mismo.
Hay que sentirse dotado para realizar alguna cosa y que esa cosa
hay que alcanzarla, cueste lo que cueste.
Marie Curie (1867-1934)

Resumen

En los últimos años el mercado de los teléfonos inteligentes o *smartphones* ha aumentado su número de usuarios, permitiéndoles que tengan acceso a internet en cualquier lugar y en cualquier momento. El problema surge cuando se quieren mantener las conexiones activas mientras el usuario se desplaza y cambia de punto de conexión.

En un principio, ninguno de los protocolos más importantes de la pila TCP/IP (IPv4, IPv6, TCP, UDP) fueron diseñados para permitir la movilidad. Por ello, aparecen diferentes soluciones en todas las capas de la pila TCP/IP que permiten mantener las conexiones activas mientras el usuario cambia de punto de conexión.

En este Trabajo Fin de Grado se presenta un estudio de las distintas soluciones de movilidad a nivel de enlace y nivel de red sobre una red real. En este estudio se mide el tiempo de traspaso o *handover* total de cada una de las soluciones, desde que el nodo móvil comienza el traspaso hasta vuelve a estar operativo recibiendo y enviando tráfico. Además se evalúan cualitativamente el funcionamiento de las soluciones.

En la primera fase, se ha desplegado la red inalámbrica en el Departamento de Ingeniería Telemática de la Universidad Carlos III de Madrid con nodos inalámbricos Saxnet Meshnode III, en la que posteriormente se ha realizado el estudio de las soluciones de movilidad.

En la segunda fase, se han estudiado soluciones de movilidad a nivel de enlace y nivel de red. A nivel de enlace, la movilidad depende de la tecnología de acceso que se utilice siendo las más utilizados Wi-Fi y la red celular (GSM, GPRS, UMTS). En este estudio se mide el rendimiento de la movilidad en una red Wi-Fi.

A nivel de red existen varios protocolos que permiten la movilidad manteniendo la dirección IP. Se dividen en dos tipos según el tipo de movilidad que ofrecen: movilidad basada en red y movilidad basada en cliente.

La movilidad basada en red es transparente al usuario y es la red quien gestiona la movilidad y la detección de movimiento de los usuarios. Proxy Mobile IPv6 (PMIPv6) es un ejemplo de este tipo de movilidad.

La movilidad basada en cliente es aquella en la que el usuario se encarga de la gestión de movilidad, participando en la señalización y en la detección del movimiento. Como ejemplo de este tipo de movilidad está Mobile IPv6 (MIPv6).

Para el estudio se han utilizando implementaciones existentes de MIPv6 y PMIPv6,

siendo necesario su instalación y configuración. En el caso de movilidad a nivel de enlace se ha diseñado una solución.

Palabras clave: IPv6, movilidad, Wi-Fi, redes celulares, MIPv6, PMIPv6.

Abstract

In recent years the market of smartphones has increased the number of users, allowing them to connect to the Internet everywhere at any time. The problem appears when they wanted to keep outgoing connections alive when the user is moving and changes his point of connection.

Initially, none of the most important protocols of the TCP/IP stack (IPv6, IPv6, TCP, UDP) was designed to allow mobility. Therefore different solutions appear in all the layers of the TCP/IP stack that allow keeping outgoing connections alive while the user is changing the point of connection.

In this Bachelor Thesis a study of different mobility solutions at link layer and network layer is presented. This study measures the handover time of each solution, since the mobile node starts the handover until it becomes operating receiving and sending traffic.

First, a wireless network has been deployed in the Telematics Department of University Carlos III of Madrid using Saxnet Meshnode III wireless nodes.

Second, different mobility solutions have been studied at link layer and network layer. At link layer, mobility depends on the access technology that has been used. Nowadays the most used technologies are Wi-Fi and cellular networks (GSM, GPRS, UMTS). In this study we measured the mobility performance of a Wi-Fi network.

At the network layer, there are several protocols to allow mobility while keeping the IP address. They are divided in two kinds of mobility: network-based mobility and client-based mobility.

Network-based mobility is transparent to the user and the network manages the mobility and movement detection of the users. Proxy Mobile (IPv6) is an example of this kind of mobility.

In client-based mobility, the client is responsible for managing mobility, taking part in the signaling and in the movement detection. An example of this kind of mobility is Mobile IPv6 (MIPv6).

In this study we used existing implementations of MIPv6 and PMIPv6, installing and configuring them. In link layer, we design a solution.

Keywords: IPv6, mobility, Wi-Fi, celullar networks, MIPv6, PMIPv6.

Índice General

Lista de Figuras

Parte I

Introducción

Capítulo 1

Introducción

1.1. Introducción

En este primer capítulo se van a describir los objetivos de este Trabajo Fin de Grado y se presentan las fases de desarrollo del mismo. Además se describe la estructura de la memoria con las partes y capítulos en los que está divida y una breve descripción de cada uno de ellos.

1.2. Objetivos

- Estudio de los diferentes tipos de movilidad y movilidad en los diferentes niveles de la pila TCP/IP.
- Estudio de la movilidad en redes celulares y redes Wi-Fi.
- Estudio de los protocolos estandarizados por el IETF (Internet Engineering Task Force), encaminados a la movilidad de un terminal.
- Despliegue de una red inalámbrica en el Departamento de Ingeniería Telemática de la Universidad Carlos III de Madrid.
- Instalación de una distribución Linux en los nodo inalámbricos y su configuración.
- Configuración de un portal cautivo en la red desplegada.
- Estudio y análisis de las implementaciones de protocolos de movilidad.
- Evaluación del tiempo total de traspaso de algunos protocolos de movilidad sobre una red real.

1.3. Fases del desarrollo

El Trabajo Fin de Grado se dividió en las fases de desarrollo siguientes:

- **Despliegue de la red**: en esta fase se desplegó la red real sobre la que luego se harían todas las pruebas. El despliegue de la red se hizo por todo el Departamento de Ingeniería Telemática de la Universidad Carlos III de Madrid.
- **Documentación y análisis del estado del arte**: en esta parte se estudiaron los protocolos de movilidad estandarizados hasta el momento, especialmente los protocolos a nivel de enlace y nivel de red.
- **Instalación y configuración de las soluciones de movilidad, desarrollando scripts de automatización**: al tener la red desplegada, se buscaron implementaciones de software libre disponibles para los protocolos MIPv6 y PMIPv6, y posteriormente, se instalaron y configuraron en los equipos necesarios según el protocolo. Además se diseñó la solución a nivel de enlace y se instalaron y configuraron los programas necesarios. Se escribieron los scripts necesarios para automatizar las pruebas y obtener un fichero con los resultados para cada protocolo.
- **Toma de medidas de cada una de las soluciones**: con los scripts programados, se procedieron a tomar las medidas de cada una de las soluciones.
- **Evaluación de los resultados**: con los resultados obtenidos, se analizaron y se dibujaron las gráficas.

1.4. Estructura de la memoria

La memoria se divide en varias partes, que a su vez se dividen en distintos capítulos. El contenido de cada parte y capítulo se resume a continuación:

1. **Primera parte: Introducción**. En esta parte se explican los objetivos, las fases del trabajo y la estructura de la memoria. Sólo contiene un capítulo:
   - Capítulo 1. Introducción
2. **Segunda parte: Estado del arte**. Se explican las diferentes soluciones de movilidad que existen en la pila TCP/IP y se analizan, con más detalle, algunos de los protocolos de nivel de enlace y nivel de red. Esta parte contiene los siguientes capítulos:
   - Capítulo 2. Movilidad
   - Capítulo 3. Protocolos de movilidad: Nivel de enlace
   - Capítulo 4. Protocolos de movilidad: Nivel de red
3. **Tercera parte: Trabajo realizado**. En esta parte se explica el trabajo desarrollado para desplegar y configurar la red inalámbrica, así como el estudio de la movilidad a nivel de enlace y nivel de red. Se divide en los siguientes capítulos:
   - Capítulo 5. Despliegue y configuración de una red inalámbrica multisalto
   - Capítulo 6. Estudio de soluciones de movilidad
4. **Cuarta parte: Conclusiones y trabajos futuros**. En esta parte se explica las conclusiones obtenidas del trabajo y futuros proyectos para ampliar el actual. Sólo contiene un capítulo:
   - Capítulo 7. Conclusiones y trabajos futuros

5. **Quinta parte: Anexos**. En esta última parte se amplía la información de algunas partes del Trabajo Fin de Grado:

   - Anexo A: Planificación de tareas y presupuesto. En este anexo se describen las tareas del proyecto y los costes de estas.
   - Anexo B: Configuración del servidor. Se presenta la configuración detallada del servidor que se utiliza para el despliegue de la red inalámbrica del departamento.
   - Anexo C: Instalación y configuración de los nodos inalámbricos. Al igual que el anexo anterior, se explica la instalación y la configuración de los nodos inalámbricos utilizados para el despliegue de la red.
   - Anexo D: Instalación y configuración de las soluciones de movilidad. Se explica la instalación y configuración necesaria para cada una de las soluciones de movilidad que se han analizado.

Parte II

Estado del Arte

Capítulo 2

Movilidad

2.1. Introducción

Hoy en día, la mayoría de personas tiene un *smartphone* o teléfono móvil inteligente con el que pueden consultar cualquier contenido desde cualquier lugar. Consultar las redes sociales y el correo electrónico, utilizar aplicaciones de chat como *Whatsapp Messenger*, etc. se han convertido en acciones normales entre usuarios. El hecho de que estos teléfonos inteligentes demandan estar siempre conectados y que es un mercado en crecimiento, hace que la movilidad sea una de las áreas que esté cobrando mayor importancia.

También hay que destacar que los usuarios utilizan principalmente dos tecnologías para conectarse: Wi-Fi y la red celular. La movilidad tiene que cumplir que sea transparente al usuario y no requiera ningún tipo de configuración cada vez que estos se desplacen. Actualmente los teléfonos son los encargados de cambiar entre diferentes tecnologías según el lugar donde se encuentren.

Además la tendencia de las redes celulares es que converjan a una red "todo-IP" (*"all-IP networks"*), con lo que ofrecer movilidad en alguno de los niveles de la pila TCP/IP supondría una ventaja para todos los usuarios.

2.2. Tipos de movilidad

La movilidad se puede clasificar de varias maneras según sus propiedades. Principalmente se diferencia entre movilidad de un usuario o movilidad de una red.

- **Movilidad de una red**: es la capacidad de una red (router de acceso y sus clientes) de cambiar el punto de acceso y seguir siendo alcanzable, enviando y recibiendo tráfico a la red móvil. Para este tipo de movilidad se definió el protocolo NEMO (Network Mobility) [DWPT05].
- **Movilidad de un cliente**: es la capacidad que tiene un cliente para cambiar su punto de acceso sin dejar de recibir y enviar paquetes dirigidos al cliente.

Además se pueden encontrar dos tipos de movilidad que aplican tanto a movilidad de una red como a movilidad de un cliente:

- **Movilidad macro**: se refiere a movilidad sobre un área amplia incluyendo todos los mecanismos de movilidad necesarios cuando un nodo móvil se mueve entre dominios.

- **Movilidad micro**: movilidad sobre un área pequeña dentro del mismo dominio. Al ser un área pequeña implica que los eventos de movilidad de los usuarios serán frecuentes, por lo que requiere que la movilidad sea lo más rápida posible.

2.2.1. Portabilidad vs. Movilidad

Normalmente se distinguen dos términos que son parecidos y que pueden llevar a confusión, son la portabilidad y la movilidad.

Portabilidad se refiere al hecho de que un usuario puede conectarse a diferentes puntos de acceso. Sin embargo, las conexiones tienen que ser reiniciadas al conectar al nuevo punto de acceso.

Movilidad se define como la capacidad de un usuario de moverse, cambiar su punto de acceso y seguir manteniendo todas sus conexiones activas, sin tener que reiniciarse las conexiones en el nuevo punto de acceso.

2.3. Niveles de movilidad

La movilidad se puede gestionar en diferentes niveles de la pila TCP/IP. Se tienen distintos mecanismos según en qué nivel se realice la gestión de la movilidad.

Figura 2.1: Pila TCP/IP

Al cambiar de red, dependiendo del nivel a que se realice la movilidad, se cambia de dirección IP, lo que implica que todas las sesiones activas hasta el momento se interrumpen y tienen que ser reiniciadas. Las sesiones se interrumpen porque se identifican a nivel de transporte por su dirección IP origen, dirección IP destino, puerto origen y puerto destino. Al moverse, cambia la dirección IP origen y ese es el motivo por el que se interrumpen.

Por esta razón se han propuesto soluciones en todos los niveles de la pila para que exista movilidad en los clientes.

2.3.1. Nivel de enlace

El nivel de enlace es el encargado del acceso al medio, de interconectar los equipos dentro de un mismo enlace y de la transmisión de datos entre ellos entre otras funciones.

Generalmente, a nivel de enlace se pueden ejecutar *handovers* o traspasos de manera rápida y *seamless*, es decir, ni el usuario ni la aplicación perciben ningún tipo de interrupción al producirse el movimiento, por lo que el usuario permanece siempre conectado. Además desde el punto de vista del nivel de red no ha producido ningún tipo de movilidad al seguir teniendo la misma dirección IP si se mueve entre puntos de acceso que pertenezcan a la misma subred. Por esta razón la movilidad a nivel de enlace es una opción atractiva.

Sin embargo tiene la desventaja que sólo es aplicable dentro de la misma tecnología de acceso, ya que cada tecnología tiene un mecanismo diferente. Por ello, se están desarrollando estándares como 802.21 [80209], estandarizado por el IEEE (*Institute of Electrical and Electronics Engineers*), o ANDSF (*Access Network Discovery and Selection Function*) [and], estandarizado por 3GPP (*3rd Generation Partnership Project*), para permitir traspasos eficientes entre diferentes tecnologías, ya que crean una capa intermedia entre el nivel de enlace y nivel de red.

En el capítulo 2 se describen los mecanismos de movilidad en redes Wi-Fi y redes celulares.

2.3.2. Nivel de red

El nivel de red o nivel IP se encarga de la transmisión de paquetes desde el origen hasta el destino. Además se encarga del enrutamiento y de mantener la calidad de servicio (QoS). Es el protocolo común en toda la pila TCP/IP y para identificar a un cliente se utiliza su dirección IP.

Al moverse un nodo móvil y cambiar de red podrían ocurrir dos opciones:

- El nodo móvil mantiene su dirección IP: implica que toda la red de Internet tendría que modificar el enrutamiento para una única dirección IP. Esta opción no es escalable ya que todos los routers de core tendrían en su tabla de reenvío rutas /32.
- El nodo móvil cambia su dirección IP: si no hay ningún mecanismo de movilidad, todas las sesiones establecidas hasta ese momento se interrumpen, por lo que el nodo móvil tendría que restablecerlas.

Para evitar que toda la red cambie su tabla de rutas o se corten todas las sesiones establecidas, el nodo móvil tiene que mantener su dirección IP y, a la vez, obtener una dirección IP en la nueva red. El IETF (*The Internet Engineering Task Force*) ha estandarizado varios protocolos que siguen este procedimiento, entre los que destacan Mobile IP (en sus versiones para IPv4 [CP02] e IPv6 [PJA11]) y Proxy Mobile IPv6 [GLD+08]. Mobile IPv6 y Proxy Mobile IPv6 se describen con más detalle en el capítulo 4.

La movilidad a nivel IP tiene ciertas ventajas sobre otras soluciones de movilidad a otros niveles:

- Permite movilidad entre diferentes tipos de redes de acceso: Wi-Fi, red celular, etc.
- Al ser una capa común en la pila, las aplicaciones no tienen que requerir ningún otro mecanismo de movilidad. Además no hay ningún cambio en la estructura de la red, por ejemplo, en la tabla de rutas de los routers de Internet.
- Al ser transparente a las capas superiores, no interrumpen las sesiones establecidas.

Sin embargo tiene una desventaja principal: para que el cliente siga siendo alcanzable, el encaminamiento se hace a través de túneles lo que implica una mayor carga y un mayor retardo.

2.3.3. Nivel de transporte

El nivel de transporte se sitúa por encima del nivel de red y se encarga de la gestión de la sesión, del establecimiento y de la liberación de recursos y del control de flujo de los datos. Los protocolos más usados en este nivel son TCP (Transmission Control Protocol)y UDP (User Datagram Protocol).

Para evitar que la conexión se interrumpa al moverse de red, se ha diseñado protocolos y soluciones a nivel de transporte que soporten este cambio:

- Stream Control Transmission Protocol (SCTP) [Ste07, RSC+02]: reemplaza a TCP, soporta múltiples direcciones IP en un cliente. Con la la extensión DAR (Dynamic Address Reconfiguration) permite añadir y borrar direcciones.
- Multipath TCP (MPTCP): proporciona extensiones a TCP para soportar funciones definidas en SCTP, sin tener que cambiar TCP en todos los nodos de la red.

La movilidad a nivel de transporte ofrece ciertas ventajas sobre la movilidad a otros niveles:

- Sin túneles: la movilidad a nivel de red se basa en túneles que ocultan la movilidad presentando siempre la misma dirección IP. La movilidad a nivel de transporte no necesita de túneles ni de enrutamiento triangular.
- Capacidad de pausar la transmisión durante el movimiento: en protocolos de capas inferiores, durante un cierto tiempo el nodo móvil no es alcanzable. Como los niveles de transporte controlan la transmisión de datos, estos pueden pausar el envío de datos hasta que se que se haya restablecido la conexión.

Sin embargo presenta otras desventajas ya que, principalmente, existen dos protocolos a este nivel, UDP y TCP, por lo que requiere soluciones para cada protocolo.

2.3.3.1. SCTP: Stream Control Transmission Protocol

SCTP [Ste07, RSC+02] define las funciones de establecimiento de la conexión, transmisión de datos, finalización de la conexión y control de congestión, por lo que es una alternativa a TCP y UDP.

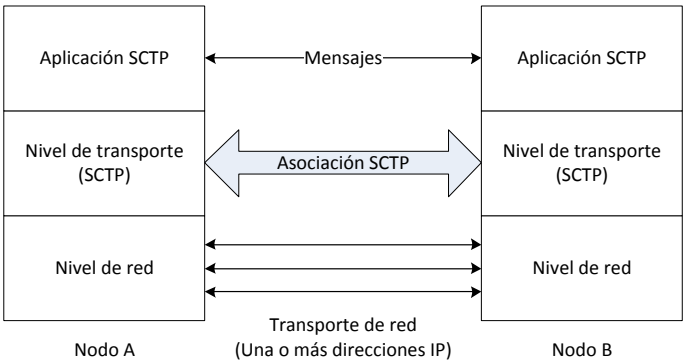

Figura 2.2: Asociación SCTP

SCTP crea asociaciones entre dos nodos que implementan SCTP. Se denomina asociación al conjunto de direcciones IP entre dos nodos. Destacan las ventajas que ofrece SCTP sobre TCP:

- Capacidad de *Multihoming*: un cliente puede utilizar varias direcciones IP durante una sesión, y tiene la opción de cambiar entre esas direcciones IP sin interrumpir la conexión.
- Capacidad de *Multistreaming*: permite tratar cada flujo de datos entre dos nodos independientemente.

Con la capacidad de *Multihoming* no se tienen la movilidad deseada, ya que no soluciona el problema de tener diferentes direcciones IP en diferentes localizaciones. Para ello se creó la extensión DAR (Dynamic Address Reconfiguration) [SXT+07], que permite añadir y borrar direcciones IP en la asociación SCTP. Al estándar SCTP junto con esta extensión se suele referir como mobile SCTP (mSCTP).

2.3.4. Nivel de aplicación

La movilidad a nivel de aplicación requiere que cada aplicación gestione la movilidad. Para ello, la aplicación es la que se tiene que encargar de restablecer la comunicación y continuar con las conexiones que la aplicación tenía abiertas.

La principal diferencia entre movilidad a nivel de aplicación y el resto de niveles es que en el resto de niveles, el identificador es la dirección IP del nodo móvil en la nueva red, mientras que en movilidad a nivel de aplicación se utiliza otro identificador, por ejemplo, un nombre se usuario.

La movilidad a nivel de aplicación tiene que ejecutar una serie de funciones:

- Autenticación: el usuario necesita crear una asociación entre el identificador de la aplicación y la dirección IP.
- Registro: el usuario tiene que registrar su dirección IP y su identificador de la aplicación con el servidor de la aplicación.
- Servicio rendezvous: un nodo corresponsal tiene que ser capaz de saber en que dirección IP está conectado el nodo móvil.

2.3.4.1. SIP mobile

Como ejemplo de movilidad a nivel de aplicación se encuentra SIP (Session Initiation Protocol) mobile.

SIP [RSC+02] proporciona la señalización necesaria para crear, modificar y finalizar sesiones. Generalmente se usa para la señalización de llamadas de VoIP (*Voice over Internet Protocol*), junto con el protocolo Session Description Protocol (SDP) para establecer los parámetros de la comunicación. Para transportar la llamada se utiliza Real Time Protocol (RTP).

En SIP se definen una serie de mensajes entre los clientes SIP (Agente de usuario, *User Agent*, UA) y los servidores Proxy, que son los encargados de reenviar los mensajes de señalización entre clientes SIP:

- REGISTER: mensaje para registrar un cliente SIP en un servidor *registrar*.
- INVITE, ACK y CANCEL: mensajes para el establecimiento de la sesión.
- BYE: mensaje para finalizar la sesión.

En la figura 2.3, se muestra un ejemplo de la señalización que se produce cuando un agente de usuario cambia de red y los mensajes que se generan.

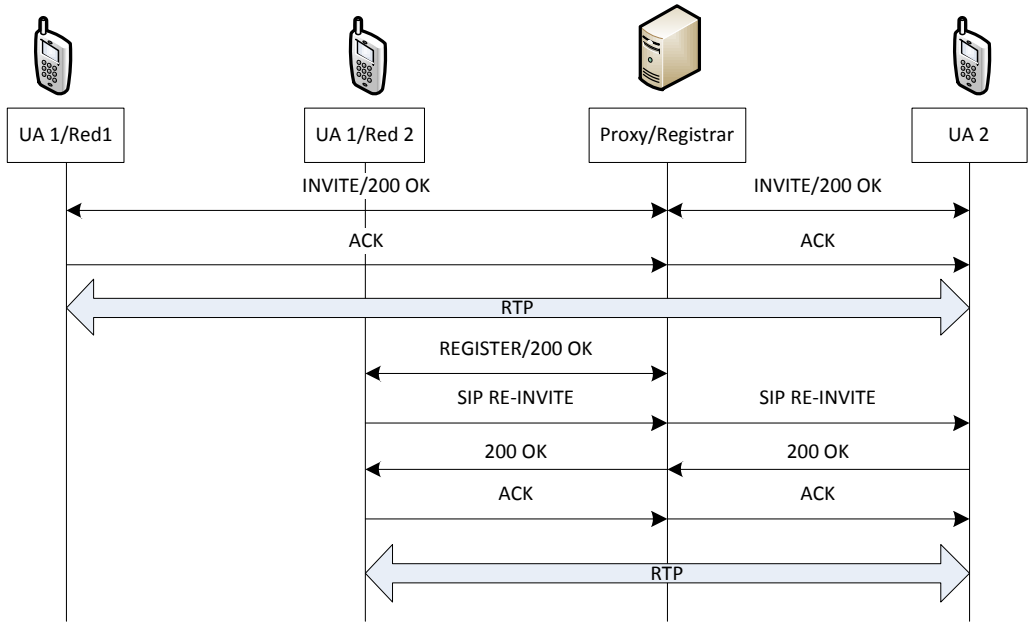

Figura 2.3: Señalización SIP

Después de establecerse la sesión y de haber comenzado la conversación, el agente de usuario se mueve. Adquiere un nueva dirección IP, registra esa IP en el servidor *registrar* y envía un nuevo INVITE con la opción "c" en la carga SDP, que contiene los datos de la conexión. El otro agente de usuario lo confirma y se vuelve a reanudar la conversación.

2.4. Conclusiones

Ya se ha visto que la movilidad es un problema, ya que los protocolos no fueron diseñados para que los usuarios se movieran. Para ello, se han diseñado soluciones a distintos niveles de la pila TCP/IP para permitir que los usuarios se muevan, seguir siendo alcanzables y no interrumpir la conexión.

Como se ha podido ver, existen múltiples protocolos que solucionan este problema. Con este Trabajo Fin de Grado se pretende estudiar tres soluciones de movilidad: una a nivel de enlace y dos a nivel de red. Se estudia cualitativamente el funcionamiento de estas y cuantitativamente el tiempo de traspaso total.

Capítulo 3

Protocolos de movilidad: Nivel de enlace

3.1. Introducción

En la movilidad a nivel de enlace el cliente tiene que detectar que su nivel de señal se ha deteriorado y decidir cambiar a otro punto de acceso que tiene un nivel de señal mejor.

Se van a describir la movilidad de las dos tecnologías más usadas para acceder a Internet y enviar y recibir tráfico: 802.11 (Wi-Fi) y la red celular.

3.2. Movilidad dentro de la misma subred IP

Hoy en día es muy común ver redes Wi-Fi desplegadas en universidades, empresas, estaciones de tren, aeropuertos, etc. Estas redes se componen de una serie de puntos de acceso y las estaciones que se van a conectar a estos.

Para entender la movilidad a este nivel, primero hay que definir los siguientes conceptos del estándar 802.11 [802]:

- Conjunto de servicio básico (*Basic Service Set*, BSS): se refiere a una red inalámbrica que tiene un punto de acceso y una serie de clientes conectados a este.
- Conjunto de servicio extendido (*Extended Service Set*, ESS): se refiere al conjunto de dos o más BSSs conectados.
- Sistema de distribución (*Distribution System*, DS): interconecta dos o más BSSs. Además reenvía los paquetes a las estaciones que están conectadas, a través de los puntos de acceso.

Para que la movilidad de la estación sea correcta, el cliente se tiene que mover a un nuevo BSS que pertenezcan al mismo ESS.

En la movilidad dentro de la misma subred IP se tiene el escenario representado en la figura 3.1. En él se tiene dos puntos de acceso (Access Point, AP) a los que se va a

conectar la estación según se vaya desplazando. Cada punto de acceso forma un BSS y están solapados entre ellos. Estos punto de acceso están conectados a un DS, con lo que los puntos de acceso pertenecen a la misma subred.

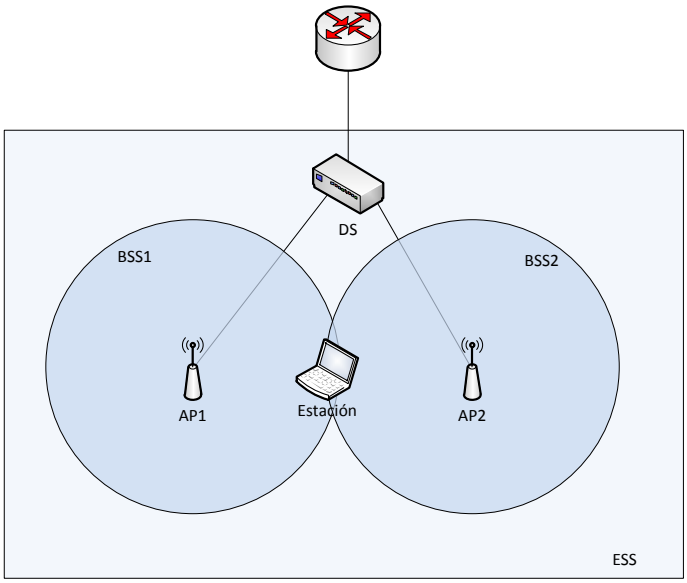

Figura 3.1: Escenario movilidad nivel de enlace en una red Wi-Fi

La estación se va a ir desplazando hasta que llegue un momento en la señal del punto de acceso AP1 sea muy baja. La estación escaneará los puntos de acceso disponibles en busca de uno que tenga una señal de potencia mayor, enviando mensajes *Probe Request* en todos los canales en busca de otro punto de acceso con el mismo identificador de red (Service Set IDentifier, SSID). Recibirá varios *Probe Request* de los puntos de accesos que estén disponibles. Seleccionará el punto de acceso AP2, que tiene un nivel de potencia mejor y el mismo SSID. Este tipo de escaneo se denomina escaneo activo. La estación también puede realizar un escaneo pasivo en el escucha los mensajes *beacon* que envía los puntos de acceso cada cierto tiempo.

Entonces se producirá la señalización en la que la estación se desasocia del AP1 y se autentica en el AP2. Al estar dentro de la misma subred, la estación mantiene la misma dirección IP y todas sus conexiones, ya que desde el punto de vista de las capas superiores no se ha producido ningún cambio.

El DS al que están conectados los puntos de acceso sigue enviando los paquetes dirigidos a la estación al AP1, ya que en su tabla de reenvío tiene una entrada que asocia la dirección MAC de la estación con la interfaz de salida del DS. Para ello, se ha diseñado Inter-Access Point Protocol (IAPP) definido en la especificación de 802.11f [20003], que permite actualizar la tabla de reenvío del DS con el envío de una trama de nivel de enlace. Además permite la comunicación entre los puntos de acceso, para que el nuevo punto de acceso informe al antiguo de que el traspaso ha terminado y que puede liberar los recursos de esa estación. 802.11f fue retirado al no existir penetración en el mercado.

Con el objetivo de que el DS reenvíe el tráfico correctamente a la estación, se puede utilizar la solución de 802.11f con el envío de una trama de nivel de enlace. Por lo que el AP2 puede enviar una trama de nivel de enlace a la dirección de broadcast con dirección origen, la dirección MAC de la estación. El DS recibe esta trama, aprende donde se encuentra la estación y reenvía el tráfico correctamente.

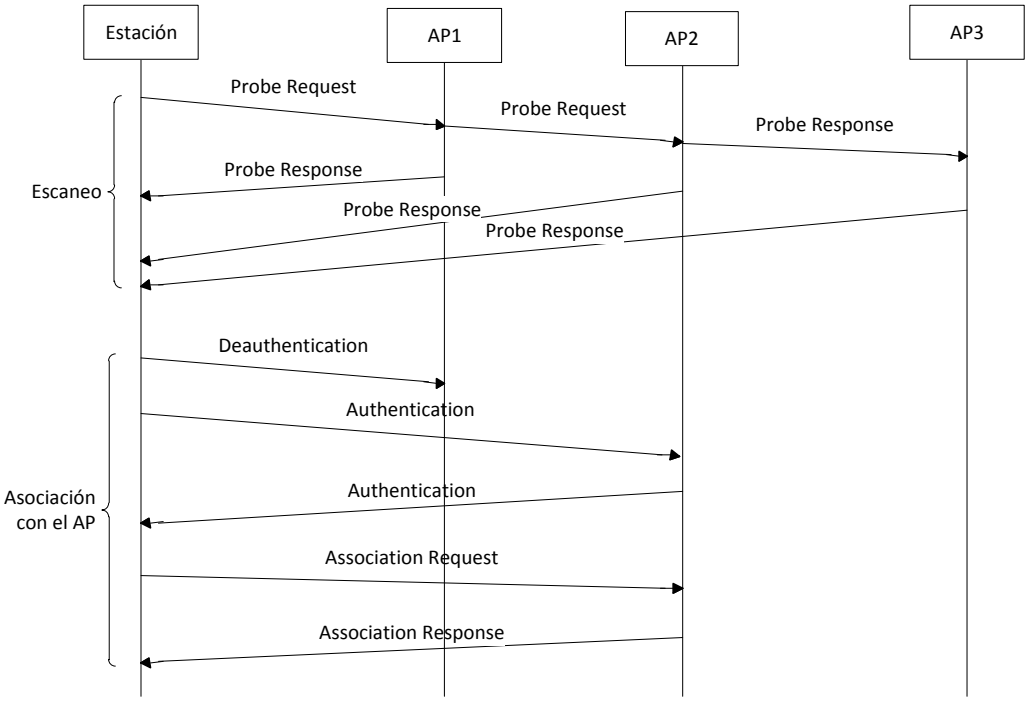

Figura 3.2: Escaneo y asociación al punto de acceso

3.3. Movilidad en redes celulares

En redes celulares se va a explicar el procedimiento de movilidad en la conmutación de paquetes (Packet Switched, PS). En primer lugar se van a explicar las entidades que participan en la movilidad:

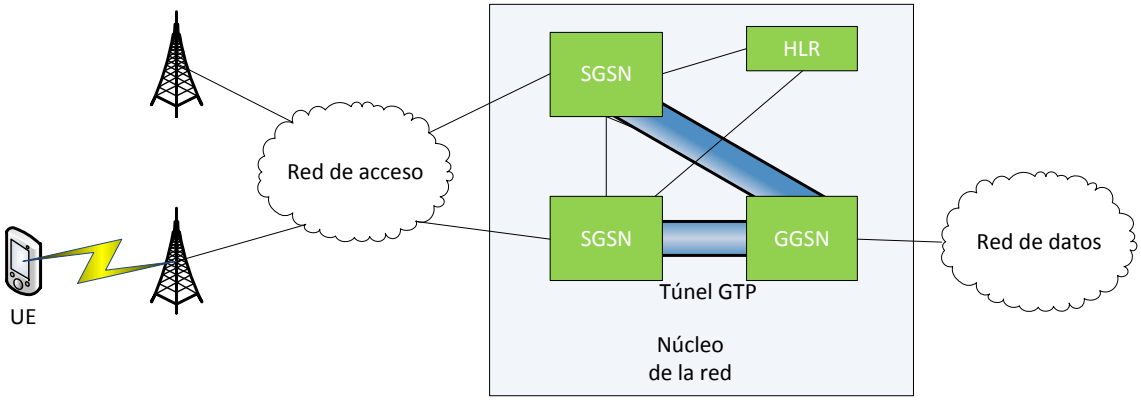

Figura 3.3: Arquitectura básica en conmutación de paquetes

- User Equipment (UE): dispositivo que va a querer comunicarse con otros dispositivos y enviar y recibir datos de la red de datos.
- Serving GPRS Support Node (SGSN): es el encargado de realizar la conmutación de paquetes junto con el GGSN. Se encarga de las acciones de enrutamiento: envía los paquetes el UE y al GGSN, según corresponda y realiza la gestión de la movilidad de los UEs.

- Gateway GPRS Support Node (GGSN): proporciona acceso a redes de datos externas al UE. Se conecta con todos los SGSNs de la red. Se encarga de enviar datos al UE a través SGSN. Para comunicarse con el SGSN correspondiente utiliza un túnel GTP (GPRS Tunneling Protocol).
- Home Location Register (HLR): es la base de datos que contiene toda la información del usuario de forma permanente.

GTP es un protocolo usado entre las comunicaciones entre el SSGN y GGSN. GTP se divide en dos planos diferentes:

- GTP-U: es el plano del usuario que permite encapsular a los datos de usuario dentro de la red de núcleo.
- GTP-C: es el plano de control. Permite la señalización entre el SGSN y GGSN para crear, eliminar y modificar los túneles.

GTP es un túnel basado en UDP que proporciona movilidad basándose en los contextos PDPs (Packet Data Protocol). Un contexto PDP se identifica por el tipo de PDP (IPv4, IPv6), la dirección IP, el perfil de calidad de servicio y el APN (Access Point Name) correspondiente. El APN corresponde a un nombre de DNS que hace referencia al GGSN.

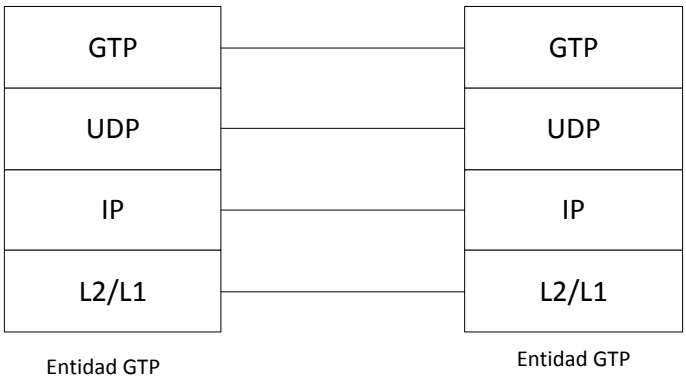

Figura 3.4: Pila del túnel GTP

Para que un dispositivo móvil puede recibir datos de Internet, primero tiene que activar un contexto PDP, el cual se activa con el procedimiento mostrado en la figura 3.5.

En él el UE envía un mensaje Activate PDP Context Request a la red. Este mensaje normalmente incluye el APN al que se debería establecer el contexto PDP.

Al recibirlo, el SGSN preguntará al DNS la dirección IP del GGSN que hace referencia en el APN. Al recibir la respuesta del DNS, enviará un mensaje Create PDP Context Request a través del plano del control de GTP, GTP-C, al GGSN.

El GGSN asignará una dirección al contexto, que puede ser IPv4 o IPv6, y contestará al SGSN con el mensaje Create PDP Context Response a través del GTP-C.

EL SGSN enviará un Radio Access Bearer Assignment Request a la red de acceso para transporta el contexto PDP. Finalmente el SGSN informará al UE de su contexto PDP con el mensaje Activate PDP Context Request Accept.

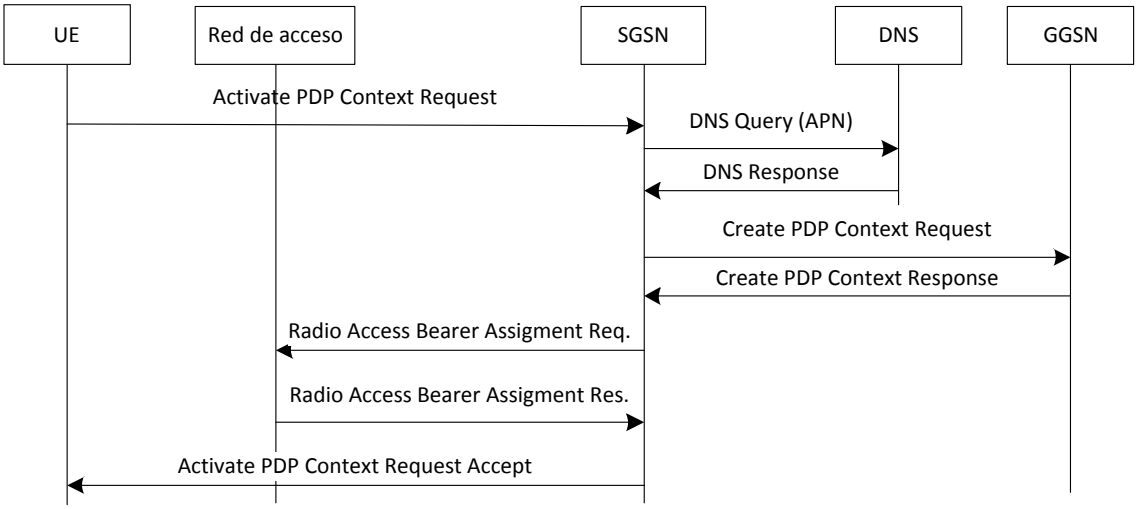

Figura 3.5: Establecimiento del contexto PDP

A partir de ese momento el UE podrá enviar y recibir tráfico de Internet. Ese tráfico entre el SGSN y el GGSN irá encapsulado en el túnel.

Si al moverse el UE y cambiar de celda, cambia a un Routing Area controlada por otro SGSN, se produce la señalización mostrada en la figura 3.6.

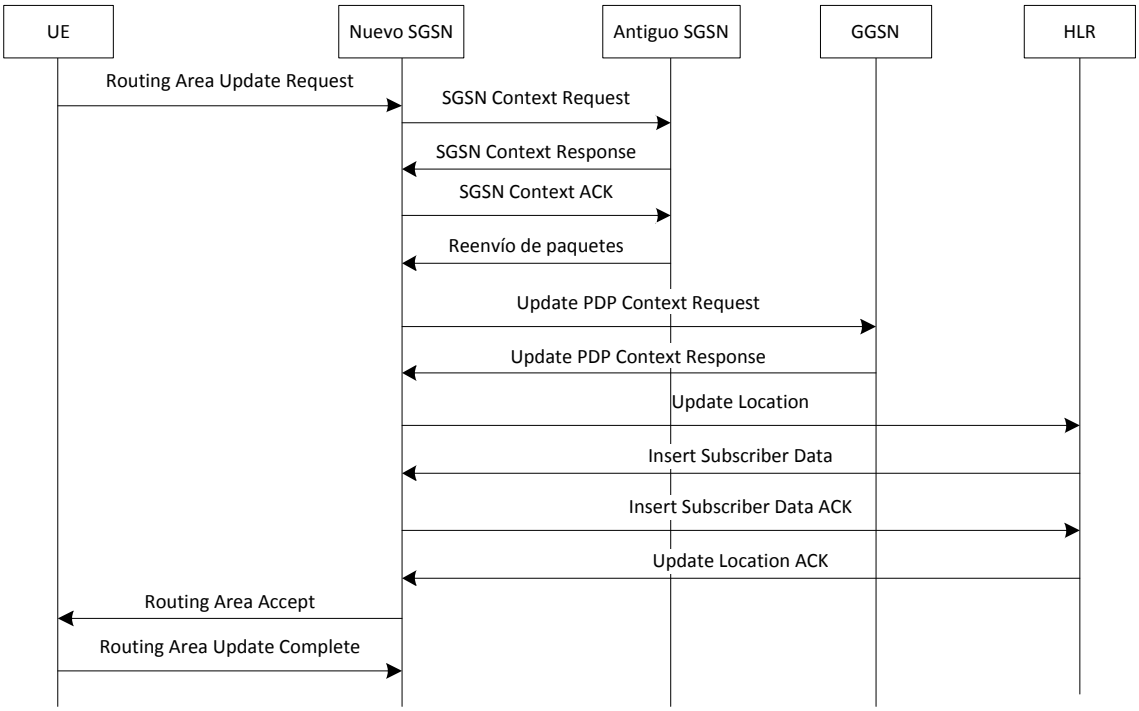

Figura 3.6: Señalización en el cambio de SGSN

El UE manda un Routing Area Update Request, que incluye el Routing Area Identifier (RAI) con la información del antiguo SGSN. El nuevo SGSN necesita recuperar la información del contexto PDP, por lo que utiliza la información incluida en el RAI para contactar con el antiguo SGSN. El nuevo SGSN envía un SGSN Context Request al antiguo SGSN a través de GTP-C.

El antiguo SGSN contesta con la información del contexto del UE a través del mensaje SGSN Context Response y que el nuevo SGSN confirmará. El antiguo SGSN reenvía los paquetes al nuevo SGSN. A su vez, el nuevo SGSN envía un Update PDP contexto Request al GGSN para que actualice la información del UE. También actualiza la localización del UE con el HLR y recibe los datos del suscriptor. Por último informa al UE que la actualización del Routing Area ha sido aceptada.

3.4. Conclusiones

Los mecanismos de movilidad a nivel de enlace son una solución de movilidad muy efectiva ya que, generalmente, son más eficientes que movilidad a niveles mayores y evita que estos tengan que ser modificados.

En este capítulo se ha explicado un mecanismo muy básico de movilidad a nivel de enlace al enviar el punto de acceso una trama de nivel dos sobre la estación. Con esto se consigue que el DS conectado a esa red aprenda la posición del usuario y reenvíe el tráfico correctamente.

En las redes celulares, el mecanismo para que el UE mantenga la dirección IP es más complejo, ya que requieren la participación de varias entidades que tienen que ser informadas cuando cambia de SGSN.

Como se ha comentado, los mecanismos de movilidad a nivel de enlace tienen la desventaja de ser únicos para cada tecnología, con lo que no permite traspasos entre tecnologías sin interrumpir las conexiones activas.

Capítulo 4

Protocolos de movilidad: Nivel de red

4.1. Introducción

En al movilidad a nivel de red, el usuario cambia de dirección IP cada vez que cambia de red. En este capítulo se van a ver dos protocolos diferentes que tienen esta característica: MIPv6 y PMIPv6.

Estos dos protocolos implementan dos tipos de movilidad diferentes:

- Movilidad basada en la red (Network-based Localized Mobility Management, NETLMM [JK07b, JK07a]): es aquella en la que la red es la encargada de detectar la localización y el movimiento del usuario; y gestionar la movilidad dentro de su dominio. PMIPv6 es un ejemplo de este tipo de movilidad.
- Movilidad basada en el cliente: es el cliente es el se encarga de la señalización y la gestión de la movilidad. MIPv6 tiene esta característica, ya que es el nodo móvil el que participa en la señalización.

4.2. MIPv6: *Mobile IPv6*

4.2.1. Introducción

El protocolo MIPv6 [PJA11, PJA04] permite a un usuario moverse de una red a otra y seguir siendo alcanzable. Para ello se define el concepto de red hogar (*home network*), que es la red a la que pertenece el prefijo de red principal del cliente. Cuando el usuario se cambia a otra red, se dice que se encuentra en una red visitada (*visited network*). El usuario detecta el cambio de red al detectar un cambio de prefijo de red.

MIPv6 tiene dos modos de funcionamiento:

- A través de la red hogar: los paquetes se encapsulan en la red visitada y se envían a la red hogar. Una vez allí, el agente local (Home Agent, HA), se encarga de reenviarlo al nodo corresponsal (Correspondent Node, CN), el nodo con el nodo móvil mantiene una comunicación.

- Directamente con el nodo corresponsal con el mecanismo de optimización de rutas (*Route Optimization*): el nodo móvil (Mobile Node, MN) se encarga de enviarlo directamente al nodo corresponsal. Para que este mecanismo funcione el nodo corresponsal tiene que implementar cierta funcionalidad de MIPv6.

4.2.2. Terminología

En el protocolo MIPv6 se definen las siguientes entidades:

- Nodo móvil (Mobile Node, MN): cliente que es capaz de cambiar de red y seguir siendo alcanzable. Participa activamente en la señalización del movimiento.
- Agente local (Home Agent, HA): se encarga de interceptar los paquete dirigidos al nodo móvil y reenviar el tráfico a este cuando se encuentra fuera de la red hogar. También se encarga de la señalización con el nodo móvil para registrar su nuevo dirección IP.
- Nodo corresponsal (Correspondent Node, CN): cliente que mantiene una comunicación con el nodo móvil. Este cliente puede ser o no un nodo móvil.

También se definen dos tipos de direcciones IP diferentes, según donde se encuentre el nodo móvil:

- Dirección hogar (Home Address, HoA): dirección IP asignada al nodo móvil que pertenece al prefijo de la red hogar. Esta es la dirección permanente y la dirección IP a la que los nodos corresponsales va a enviar el tráfico, si no hay ningún tipo de mecanismo de optimización de rutas.
- Dirección red visitada (Care-of Address, CoA): dirección IP del nodo móvil cuando se encuentra en una red visitada, que registrará al agente local como su dirección de contacto. Esta dirección va variando según el nodo móvil cambie de red.

4.2.3. Procedimiento básico

MIPv6 está diseñado para que la movilidad sea transparente para las capas superiores y para el resto de Internet. Para ello se usa la dirección hogar el nodo móvil: todo el tráfico de los nodos corresponsales van a ir dirigidos a esta dirección y las capas superiores van a ver siempre la misma dirección IP.

Mientras el nodo móvil se encuentre en la red hogar, su procedimiento será el normal: los paquetes irán dirigidos a la dirección hogar, que pertenece al rango de la red hogar.

Cuando el nodo móvil se mueve desde su red hogar a una red visitada, configura su nueva dirección IP, la dirección CoA. Esta dirección se puede configurar basados en mecanismos de autoconfiguración sin estado (*stateless*) o mecanismos con estado (*stateful*), como por ejemplo DHCPv6. Para que la detección de movimiento sea más rápida, hay dos opciones:

- El nodo móvil puede utilizar los mecanismos de Router Discovery y Neighbor Unreachability Detection (NUD) para verificar el enlace en el que se encuentran. También puede utilizar la información recibida a nivel de enlace.

- Puede adoptar una actitud más pasiva y escuchar los mensajes Router Advertisement (RA) que el router de acceso (Access Router, AR) envía cada cierto tiempo. Este método tiene la desventaja que el nodo móvil puede tardar varios segundos en darse cuenta que se ha movido de red, según la configuración que tenga el nuevo router de acceso. Para que este tiempo sea lo menor posible en la RFC 6275 [PJA11] se rebajan estos tiempos.

Después de configurar su dirección CoA, el nodo móvil informa de esta nueva dirección al agente local con un mensaje Binding Update (BU). Este mensaje contiene la asociación entre la dirección hogar, HoA; y la dirección de la red visitada, CoA.

El agente local registra esta nueva entrada en su tabla de asociaciones (Binding Cache), donde tiene almacenadas todas las asociaciones de los nodos móviles a los que sirve. Si el BU es aceptado, el agente local busca si existe alguna entrada para este nodo, si es así la actualiza y sino crea una nueva. El agente local envía el mensaje Binding Acknowledgement (BA) para confirmar la asociación.

El agente local suplanta al nodo móvil dentro de la red hogar. Por ello, para asegurarse que todo el tráfico llega el agente local, este envía un mensaje Proxy Neighbor Advertisement a la dirección multicast de todos los nodos, y ejecuta el mecanismo detección de dirección duplicada, (Duplicate Address Detection, DAD), para la dirección hogar del nodo móvil.

A partir de ese momento el agente local recibe todo el tráfico dirigido al nodo móvil. El agente local lo reenvía encapsulado a través de un túnel bidireccional.

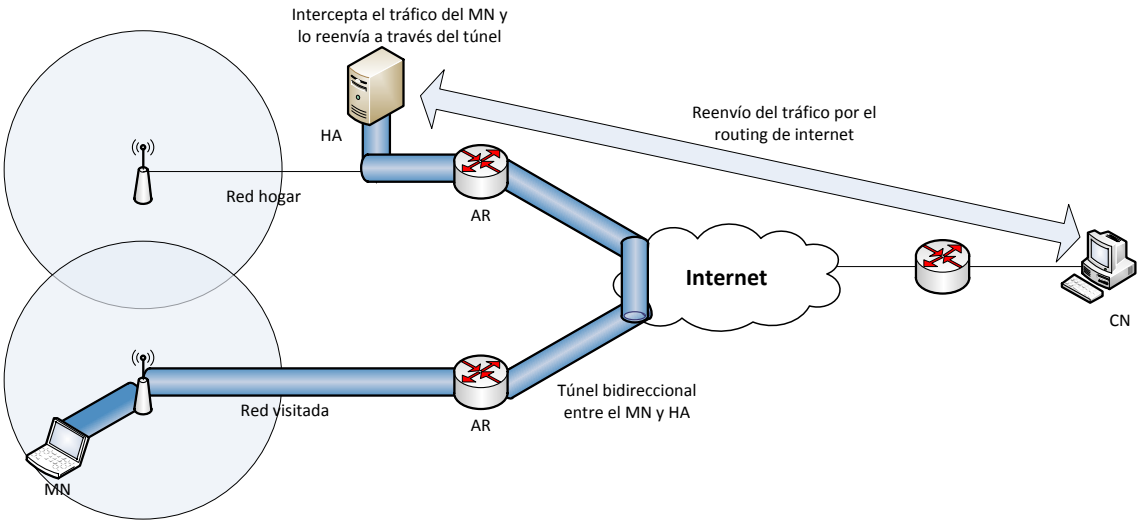

Figura 4.1: Funcionamiento básico MIPv6

- Si el paquete va dirigido del nodo corresponsal al nodo móvil, el agente local lo intercepta, lo encapsula y lo reenvía añadiendo una nueva cabecera IPv6, en la que la dirección IP de origen es la del agente hogar y la dirección destino, la CoA.
- Si el paquete va dirigido del nodo móvil al nodo corresponsal, el nodo móvil envía un paquete con dos cabeceras: la primera con dirección origen su CoA, y dirección destino al del agente local; y la segundo con dirección origen su HoA y dirección destino la del nodo corresponsal. Al interceptar el paquete el agente local, desencapsulará el

paquete quitando la primer cabecera y lo reenviará al nodo corresponsal. En la figura 4.2 se puede ver este comportamiento.

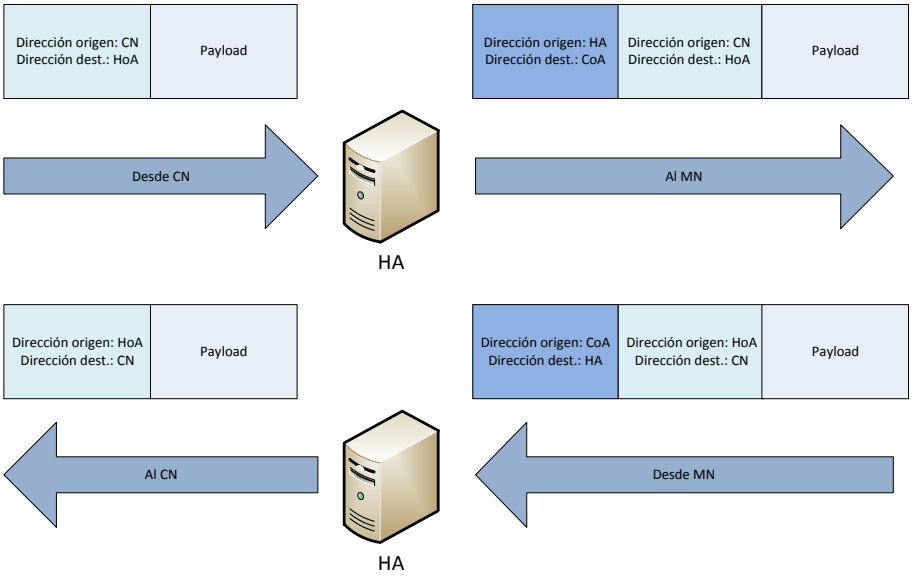

Figura 4.2: Reenvío agente local

Este modo de funcionamiento tiene la desventaja de que todo el tráfico hay que reenviarlo al agente local por un túnel, con lo que se produce un mayor retardo en la comunicación y más carga en el paquete al tener que introducir otra cabecera. Para ello MIPv6 tiene la posibilidad de optimizar rutas (*Route Optimization*), en el que el nodo móvil anuncia al nodo corresponsal su dirección CoA.

4.3. PMIPv6: *Proxy Mobile IPv6*

4.3.1. Introducción

Proxy Mobile IPv6 (PMIPv6) [GLD+08], estandarizado en la RFC 5213, es un protocolo de movilidad basado en la red, con lo que se permite la movilidad de un cliente sin que cambie su dirección IP dentro del dominio Proxy Mobile IPv6 (Proxy Mobile IPv6 Domain, PMIPv6-Domain).

Está diseñado para que se pueda utilizar con diferentes tecnologías de acceso como Wi-Fi, WiMAX, 3GPP, etc. PMIPv6 se basa en MIPv6 ya que reutilizar la funcionalidad del agente local y el formato de mensajes utilizados en la señalización.

4.3.2. Terminología

En PMIPv6 se definen dos entidades:

- Mobile Access Gateway (MAG): es el router de acceso que se encarga de la detección de movimiento de un nodo móvil dentro de su dominio. También se encarga de encapsular y desencapular los paquetes con origen o destino del nodo móvil. En un dominio PMIPv6 hay varios MAGs.

- Local Mobility Anchor (LMA): se conecta a todos los MAGs del dominio. Mantiene una lista con las rutas hacia los nodos móviles conectados al dominio PMIPv6. Reenvía todo el tráfico enviado desde el nodo móvil o enviado a este a través de un túnel bidireccional entre el LMA y el MAG correspondiente.

4.3.3. Procedimiento básico

Cuando un nodo móvil entra en un dominio PMIPv6, se conecta a uno de MAGs. El nodo móvil envía un mensaje Router Solicitation (RS) para descubrir los routers del enlace y autoconfigurar su dirección IP. El MAG verifica la identidad del nodo móvil conectado y comprueba si el nodo móvil está autorizado para entrar en el dominio. Además obtiene el un identificador de nodo móvil en la red (Mobile Node Identifier, MN-ID), con el que el LMA podrá saber si ese nodo móvil se acaba de conectar o si ha cambiado de MAG. Este identificador tiene que ser único para cada nodo móvil en el dominio. Si la verificación del

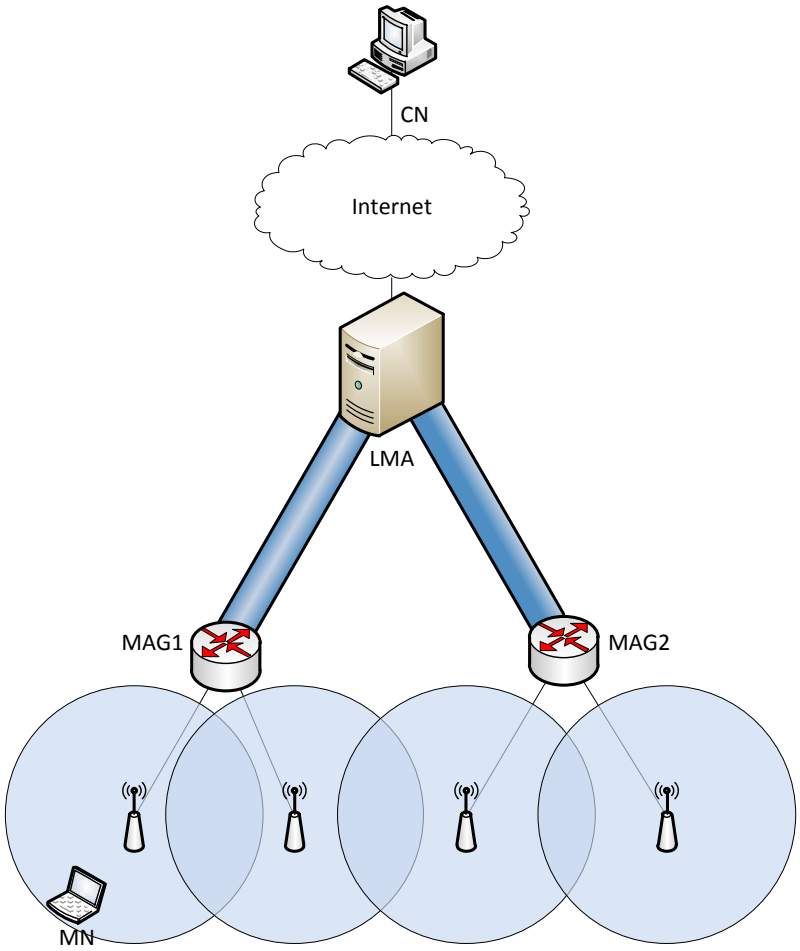

Figura 4.3: Diagrama de red de PMIPv6

nodo móvil es correcta, el MAG comienza el registro del nodo móvil con el LMA enviando un mensaje Proxy Binding Update (PBU). Este mensaje se incluye el identificador del nodo móvil, así como el identificador del MAG, Proxy-CoA, que es la dirección IP de salida del MAG.

El LMA recibe el paquete PBU del MAG y verifica si el mensaje es correcto con la

especificación. Si es así, el LMA comprueba si ya existe una entrada para ese identificador de nodo móvil en su tabla Binding Cache. Si no existiese ninguna entrada, asigna un uno o más prefijos de red al nodo móvil. Estos prefijos se denominan Home Network Prefixes y al menos alguno de ellos se asigna al nodo móvil.

El LMA crea una nueva entrada en su tabla (Binding Cache Entry, BCE), en la que se especifica el identificador del nodo móvil, la dirección Proxy-CoA del MAG y los prefijos que se han asignado al nodo móvil. Además crea un túnel bidireccional con el LMA que envió el PBU, si no existiese, y configura la ruta para los prefijos asignados al nodo móvil a través del túnel. Por último, el LMA crea el mensaje Proxy Binding Acknowledgement (PBA), que incluye el identificador del nodo móvil y los prefijos asignados.

Al recibir el PBA, el MAG configura el túnel bidireccional, si no existiera, y al igual que el LMA, configura las rutas a través del túnel. A continuación envía un Router Adverstisement (RA), al nodo móvil con el prefijo asignado. Con este información el nodo móvil puede configurar su dirección IP con el mecanismo de autoconfiguración sin estado y enviar y recibir tráfico de Internet.

Mientras el nodo móvil esté dentro del dominio PMIPv6, todo el tráfico dirigido al nodo móvil es recibido por el LMA. Este encapsula el tráfico añadiendo una cabecera en el que la dirección IP origen será la IP del LMA, mientras que la IP destino será la Proxy-CoA del MAG; y lo reenvía al MAG correspondiente. El MAG desencapsulará el paquete, eliminando la cabecera añadida por el LMA y lo reenviará al nodo móvil.

El MAG es el router por defecto del nodo móvil y enviará todo el tráfico a este. El MAG recibe un paquete del nodo móvil, lo encapsula en el túnel añadiendo una cabecera en el que la dirección IP de origen será la Proxy-CoA del MAG, y la IP destino, la dirección IP del LMA. El LMA recibe el paquete, elimina la cabecera añadida por el MAG y lo reenvía al nodo corresponsal.

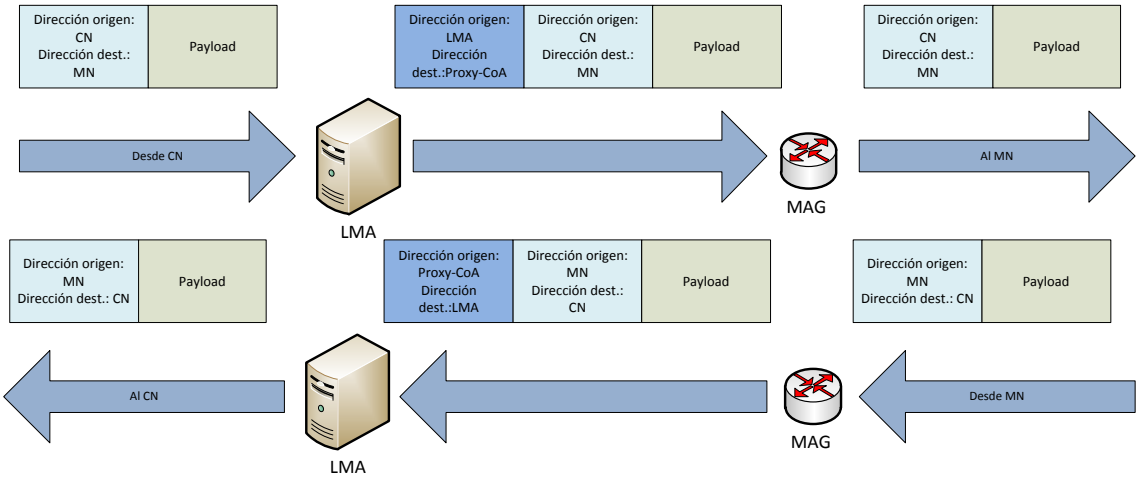

Figura 4.4: Reenvío en PMIPv6

4.3.3.1. Cambio de MAG

Después de obtener la configuración inicial, el nodo móvil puede cambiar de punto de acceso y puede que de MAG, según la arquitectura de la red. Para que el nodo móvil siga estando conectado, el nuevo MAG tiene que detectar que un nodo móvil se ha conectado;

y el antiguo MAG tiene que detectar que el nodo móvil ya no es alcanzable. Para ello se puede escuchar eventos de nivel de enlace o eventos Neighbor Unreachability Detection (NUD).

Cuando el antiguo MAG detecta que el nodo móvil se ha desconectado, envía un PBU al LMA para eliminar el registro del nodo móvil. Después de un tiempo o al recibir la confirmación, el MAG eliminará todas las rutas configuradas para ese nodo móvil.

El LMA al recibir el PBU, envía un PBA al antiguo MAG y lanza un temporizador. Durante ese tiempo el LMA tira todos los paquete dirigidos al nodo móvil. El LMA espera recibir un PBU del nuevo MAG, actualizando la localización del nodo móvil. Si no se recibe ningún PBU, el LMA eliminará la entrada del nodo móvil en su tabla Binding Cache.

Cuando el nuevo MAG detecte el nodo móvil conectado, comenzará la señalización con el mismo procedimiento que cuando se conecta un nuevo nodo móvil: envía un PBU al LMA, con el identificador del nodo móvil y la dirección Proxy-CoA del MAG. El LMA recibirá el mensaje y comprueba que ya existe una entrada en su tabla Binding Cache para ese nodo móvil. Actualiza la entrada y actualiza el túnel y las rutas correspondientes. Por último envía un PBA al nuevo MAG, con los prefijos asignados anteriormente al nodo móvil. A partir de ese momento el MAG enviará RAs con los prefijos asignados al nodo móvil. Con este procedimiento se asegura que, desde el punto de vista del nivel de red del nodo móvil, no se ha producido ningún tipo de movilidad, ya que siempre se anuncia el mismo prefijo.

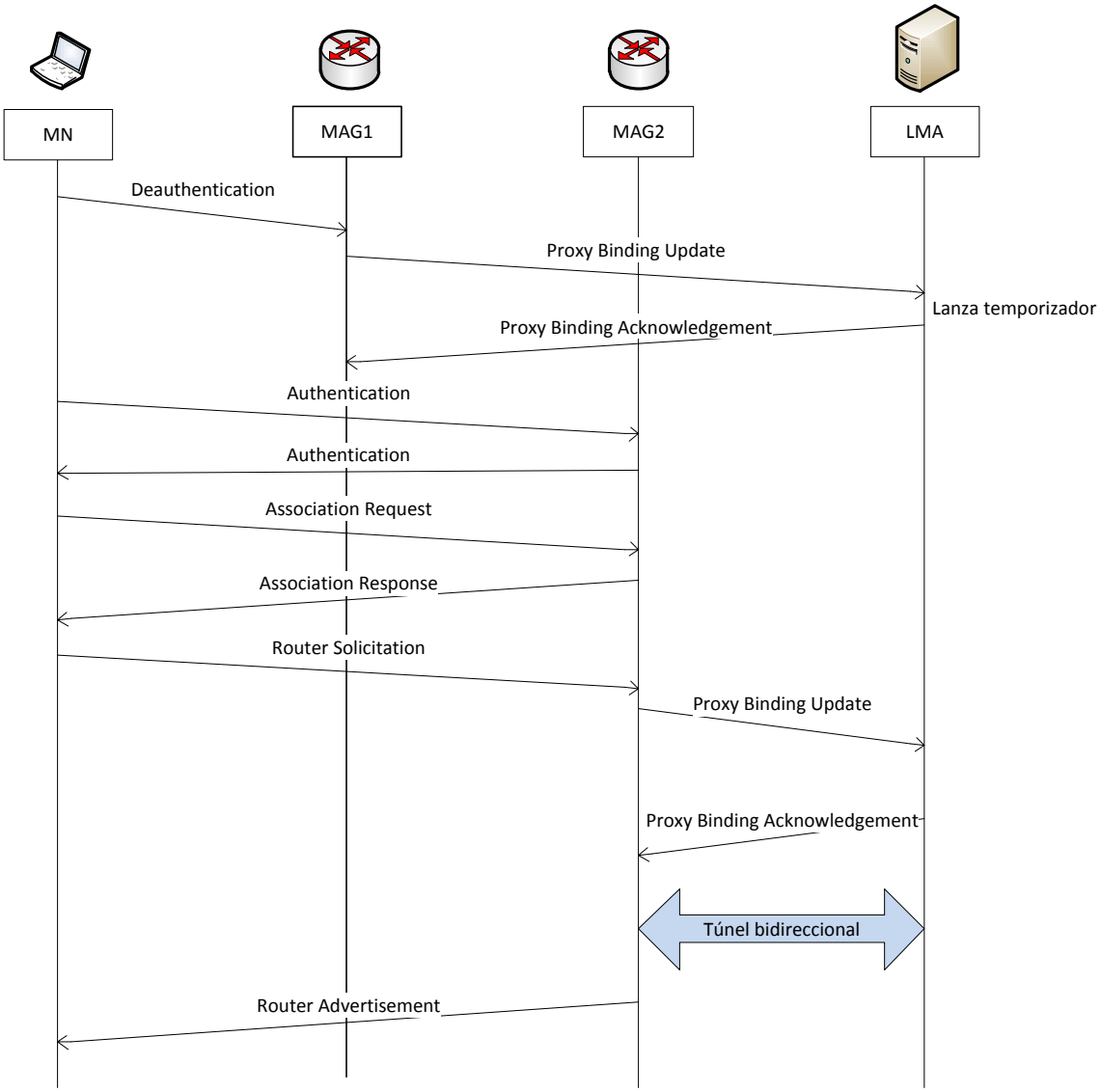

Figura 4.5: Diagrama de secuencia: procedimiento de cambio de MAG

4.4. Conclusiones

La movilidad a nivel IP requiere de túneles lo que conlleva un mayor retardo en la transmisión de los datos. Además hay que enviar la señalización necesaria para informar de la localización del nodo móvil.

En el caso de MIPv6, además de estos inconvenientes, tiene un mayor retardo al tener que enviar todos los datos al agente local y este al nodo corresponsal, si no se ha ejecutado la optimización de rutas. Para que este retardo sea menor se han estandarizado protocolos como HMIPv6 (Hierarchical Mobile IPv6) [SCEB08], que sus principales objetivos son reducir la señalización, mejorar el retardo el moverse y ejecutar la optimización de rutas sin señalización.

En PMIPv6 todo el tráfico enviado por un nodo móvil va a atravesar el dominio hasta llegar al LMA, y el LMA será el encargado de reenviar el tráfico hasta el nodo corresponsal. En el caso en el que el nodo corresponsal se encuentre en el mismo dominio, el tráfico seguirá

siendo reenviado al LMA, cuando se podría reenviar entre MAGs [LJW11].

Además de este problema se daría otro caso en el que un nodo móvil que implemente MIPv6 entre en un dominio PMIPv6, se produciría una gran carga ya que en el enlace entre el LMA y el MAG habría tres cabeceras IPv6.

Como se observa, en este área existen varios problemas que todavía están por resolver, por lo que es una de las áreas con más futuro.

Parte III

Descripción del trabajo realizado

Capítulo 5

Despliegue y configuración de una red inalámbrica multisalto

5.1. Introducción

En este capítulo se va a describir el despliegue y configuración de una red inalámbrica multisalto en el Departamento de Ingeniería Telemática de la Universidad Carlos III de Madrid. Esta red va a tener dos puntos de acceso diferentes: uno protegido por un usuario y contraseña y otro abierto, en el que para poder navegar por Internet hay que aceptar unos términos y condiciones.

Se va a comentar los equipos utilizados, la estructura de la red, la configuración necesaria en los diferentes equipos y el despliegue de los nodos por todo el departamento. Esta parte del trabajo se realizó junto a Ricardo Martínez Barrero, como parte de su Trabajo Fin de Grado.

5.2. Objetivos

Para desplegar la red inalámbrica se utilizaron dos tipos de equipos:

- Nodos inalámbricos: ofrecen acceso Wi-Fi a los usuarios. Se conectan a otros nodos a través de enlaces inalámbricos y al servidor a través de la red cableada.
- Servidor: es el controlador de los nodos inalámbricos y el que proporciona acceso a Internet tanto a los nodos inalámbricos como a los usuarios que se conecten a estos.

Estos equipos se interconectan como aparece en la figura 5.1.

El objetivo que se perseguía al realizar el despliegue es la creación de una red inalámbrica mallada o una red *mesh*, en la que todos los nodos inalámbricos estuvieran conectados entre ellos sin necesidad de estar conectados por un cable. Por lo que los nodos no sólo reciben tráfico de las estaciones conectadas o del servidor, sino que también actúan como router para el restos de nodos.

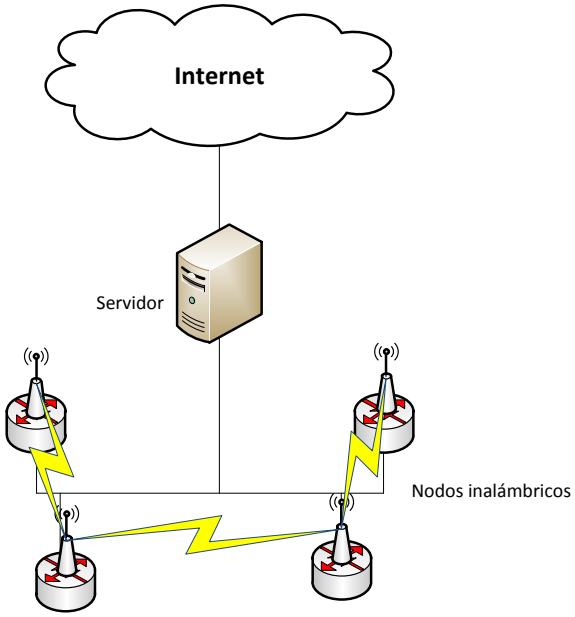

Figura 5.1: Arquitectura de la red desplegada

5.3. Equipos utilizados

5.3.1. Servidor

El servidor es el elemento principal de la red ya que recibe todo el tráfico generado de los usuarios y por los nodos, y se encarga de reenviarlo al resto de Internet. Tiene instalado un portal cautivo para obligar a los usuarios a autenticarse para poder dar acceso por Internet, así como todo el software necesario para la gestión de la red y de los nodos, que se explicará a continuación. Se utilizó el equipo *escorpion* que se encuentra la sala de servidores 4.1A01.

5.3.2. Nodos inalámbricos

Se han utilizado los equipos Saxnet Meshnode III como nodos inalámbricos. Estos equipos disponen de cuatro tarjetas inalámbricas Atheros 802.11 a/b/g, con las que se puede crear los puntos de acceso y los enlaces para comunicarse con otros nodos. Asimismo tienen instalada la distribución Debian, que ofrece una mayor versatilidad en la configuración.

5.4. Configuración del servidor

El servidor es el elemento central de la red ya que realiza las siguientes funciones:

- Da acceso a Internet a los nodos y a los usuarios conectados. Para ello se emplea NAT y utiliza su interfaz con dirección pública para comunicarse con el resto de Internet.

- Gestiona un portal cautivo por el que todos los usuarios tendrán que autenticarse para tener acceso a Internet.
- Se encarga de consultar el estado de cada uno de los nodos inalámbricos y de notificar el fallo de alguno de estos.

En las siguientes secciones se va a explicar la configuración de cada una de las funciones. Para una descripción más detallada de la configuración del servidor, se puede consultar el anexo A.

5.4.1. Acceso a Internet

Para dar acceso a Internet a los nodos inalámbricos y a los usuarios que se conectaran se decidió crear una serie de reglas con las herramienta `iptables`. Se configuró para que el servidor hiciera de router y creara una NAT. Con lo que los nodos inalámbricos y los usuarios tendrían una IP privada y se comunicarían con el resto de Internet con la IP pública del servidor.

5.4.2. Configuración del portal cautivo

5.4.2.1. Introducción

Un portal cautivo es una aplicación que controla el acceso de usuarios a una red. Para ello bloquea al tráfico de los usuarios redireccionándolos a una página web y obligándoles a autenticarse para que puedan navegar por Internet. Un ejemplo de portal cautivo lo tenemos en la universidad con la red WiFi-UC3M, en la que no se puede navegar hasta que se abra el navegador y se acepten los términos y condiciones.

Para la configuración del portal cautivo, primero se hizo una investigación de los programas que ofrecían este tipos de soluciones y ver la que mejor cubría nuestras necesidades. Algunos se descartaron porque ofrecían una distribución Linux que era un portal cautivo, pero no una aplicación que se instalara en una distribución. Al final se hizo una selección de todas ellas y nos quedamos con dos:

- Chillispot: aplicación de software libre disponible para distribuciones Linux.
- Pepperspot: aplicación derivada de Chillispot que además ofrece soporte de IPv4 e IPv6.

Se utilizó la aplicación Chillispot ya que, aunque Pepperspot era un derivado y ofrecía más opciones que Chillispot, al probar la aplicación se observó que no funciona correctamente sobre IPv4.

5.4.2.2. Características de Chillispot

Para la configuración del portal cautivo con el programa Chillispot se necesita de:

- Conexión a Internet

- Puntos de acceso inalámbricos
- Un servidor Radius
- Un servidor web

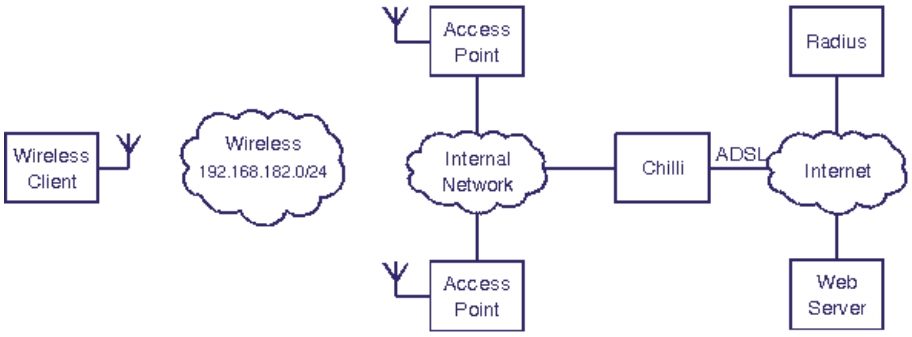

Figura 5.2: Escenario de Chillispot. Fuente: [chi]

Chilli es el software que se instala en el PC. Ofrece el tipo de autenticación Universal Access Method (UAM), que al conectarse un usuario solicitar una dirección IP y chilli se encarga de asignarle una. Cuando el usuario abra su navegador, chilli capturará su sesión TCP y le redirigirá a la página de autenticación segura. Cuando el usuario se autentique, podrá navegar por Internet. Para autenticar al usuario, chilli reenvía las peticiones al servidor Radius.

5.4.2.3. Configuración de Chillispot

Como se observa en la figura 5.2, obtenida de la página web oficial, los puntos de acceso tienen que estar directamente conectados al servidor Chilli.

El programa Chilli exige que los puntos de acceso tienen que estar a un salto de éste. El escenario en el que se iba a montar la red, iba a existir una red Ethernet en medio, por lo que se configuró una red privada virtual (VPN), para evitar este punto.

Para crear la VPN se utilizó el programa OpenVPN en modo *Ethernet Bridging*, para simular una red Ethernet. Para ello, hubo que crear un *bridge* entre la interfaz que crea el programa OpenVPN y la interfaz del servidor. Este *bridge* no se podía crear con la interfaz que daba conexión a Internet al equipo, ya que al crear el *bridge*, elimina su dirección IP. Por lo que fue necesario crear una interfaz virtual con el módulo *dummy*. Con esta configuración se tiene simulada la red de forma que se elimina la restricción del Chillispot.

Además, como se observa en la figura anterior 5.2, Chilli necesita otras dos aplicaciones para funcionar ambas instaladas en el servidor:

- Un servidor web, para mostrar la página web en la que el usuario se va a autenticar. Se utilizó el programa Apache que se instaló en el mismo equipo.
- Un servidor Radius para la autenticación de los usuarios. Se utilizó el programa FreeRADIUS que se instaló en el mismo equipo.

Finalmente se configuró el servidor, indicado en los ficheros `chilli_auth.conf` y `chilli_no_auth.conf`, en los que había que especificar varias opciones entre ellas:

- La interfaz del servidor de DHCP y su rango de direcciones.
- Servidores DNS para la configuración del cliente.
- Dirección IP del servidor Radius y su clave *shared secret*.
- Página web para la redirección del usuario.

Asimismo como se configuró para tener dos SSIDs diferentes, uno abierto y accesible aceptando los términos y condiciones, y otro privado con un nombre de usuario y contraseña; fue necesario lanzar el portal cautivo y el OpenVPN dos veces con dos ficheros de configuración distintos.

5.4.3. Nagios

Nagios es un programa de software libre para monitorizar equipos, servicios y aplicaciones enviando alertas en caso de que se produzca algún fallo en algunos de estos.

En nuestro caso se usó para verificar el estado de los nodos inalámbricos y recibir en correo cuando el estado de estos cambiase. Para ello se programó un ping cada cierto tiempo. En caso de que alguno de los nodos no contestara, automáticamente enviaba un correo electrónico identificando al nodo.

Como se observa en la figura 5.3, el programa dispone de una interfaz web con la que se puede saber el estado de los nodos, los servicios asociados a cada uno, y enviar ciertos comandos a los equipos para que ejecuten, entre otras opciones.

Figura 5.3: Página web Nagios

5.5. Configuración de los nodos inalámbricos

A continuación se va a comentar la configuración de los nodos. En el anexo B se puede consultar la instalación y la configuración detallada de los nodos inalámbricos.

5.5.1. Instalación

En los equipos Saxnet Meshnode III tenían instalado por defecto una distribución específica del fabricante basada en Debian 4 'Etch' y el kernel 2.6.24. Se quería tener una instalación personalizada (distribución, kernel, software, etc.), para no depender del software del fabricante pero también mantener la instalación incluida por defecto. Para ello, creamos una partición en la memoria principal de los nodos para instalar la distribución Linux. Realizamos la instalación de la versión Debian Squeeze 6 e instalamos un kernel personalizado. La versión de kernel que instalamos fue la 2.6.39.2 compilada con las mismas opciones que incluía el fichero de configuración del kernel por defecto, añadiendo las opciones del kernel del driver ath5k.

5.5.2. Drivers de las tarjetas inalámbricas

Además de ath5k, compilamos e instalamos Madwifi. Madwifi es el driver de tarjetas inalámbricas que más tiempo lleva desarrollado en Linux para el *hardware* Atheros, mientras que ath5k es más nuevo y el sustituto natural de Madwifi a largo plazo. Aunque Madwifi debería dar menos fallos que ath5k al estar más tiempo disponible y, por tanto, más desarrollado y probado; hacía que los nodos fueran inestables al mandar tráfico, ya que dejaban de responder al enviar más datos de los que era capaz de procesar. Por ello, realizamos la migración a ath5k, consiguiendo una mayor estabilidad y el mismo ancho de banda.

Para configurar el driver ath5k hubo que instalar los siguientes paquetes:

- `iw`: comando que permite crear interfaces a partir de la tarjeta, cambiar la frecuencia, la potencia, la antena por la que se emite, etc.
- `hostapd`: programa para crear puntos de acceso.
- CRDA (Central Regulatory Domain Agent): paquete que configura los canalas de las interfaces inalámbricas según la región.

Además de este último paquete fue necesario aplicar un parche[1] para que estuvieran permitidos todos los canales según el país, tanto en la banda de 5GHz como en la de 2.4GHz, ya que no se configuraban correctamente.

5.5.3. Configuración de la memoria principal

Los nodos inalámbricos incluyen una memoria flash como disco duro principal. Al no poderse cambiar, tener unos ciclos limitados de escritura y ser un despliegue a largo plazo,

[1] https://dev.openwrt.org/browser/trunk/package/mac80211/patches/403-ath_regd_optional.patch

instalamos una memoria USB en la que se almacenan las carpetas en las que el sistema operativo tiene que escribir: `/var` y `/tmp`. Además el resto de la memoria se configura en modo sólo lectura para que no se produzcan escrituras innecesarias, pudiendo cambiar a modo lectura y escritura en cualquier momento.

5.5.4. OpenVPN

Finalmente se añadió el software necesario para crear la red virtual privada y puentear la interfaz inalámbrica con la interfaz que crea el OpenVPN, para que los puntos de acceso estuvieran directamente conectados con el portal cautivo. Además, como se ha comentado que existen dos redes diferentes, se crearon los dos puntos de acceso usando puntos de accesos virtuales (Virtual Access Point, VAP) y se configuraron las dos VPNs.

5.6. Despliegue de la red

5.6.1. Introducción

El despliegue de la red se realizó por distintos despachos y laboratorios del departamento de Ingeniería Telemática de la Universidad Carlos III. Los despachos se encontraban en dos redes diferentes, algunos estaban conectados a la red 163.117.139.0/24 y otros se conectaban a la red 163.117.140.0/24, por lo que fue necesario que el servidor estuviera conectado a las dos subredes y que los nodos tuvieran direcciones IP diferentes según en la red que se encontraran. En la siguiente sección se especifica la estructura y la configuración de la red con el portal cautivo.

5.6.2. Estructura de la red

Al estar en subredes diferentes se tomó la decisión de asignar direcciones IP diferentes según en la red que se encontraran. A los nodos que se encontraban conectados a la red 163.117.139.0/24 tenían direcciones IP del rango 10.0.139.0/24, mientras los que se encontraban conectados a la red 163.117.140.0/24 tenían las direcciones IP 10.0.140.0/24. Además fue necesario añadir direcciones IP al servidor dentro de esos rangos en las interfaces de cada una de las subredes para conectarse a los nodos. En la figura 5.4 se observa la configuración de la red.
Como se ha comentado anteriormente, en las interfaces *dummy* del servidor se crean los túneles OpenVPN.

5.6.3. Colocación de los nodos

En total se desplegaron ocho nodos por todos los despachos del departamento, dejando otro más configurado en caso de que alguno de los desplegados fallara. Se colocaron en los siguientes despachos:

- meshnode01: 4.0.F15

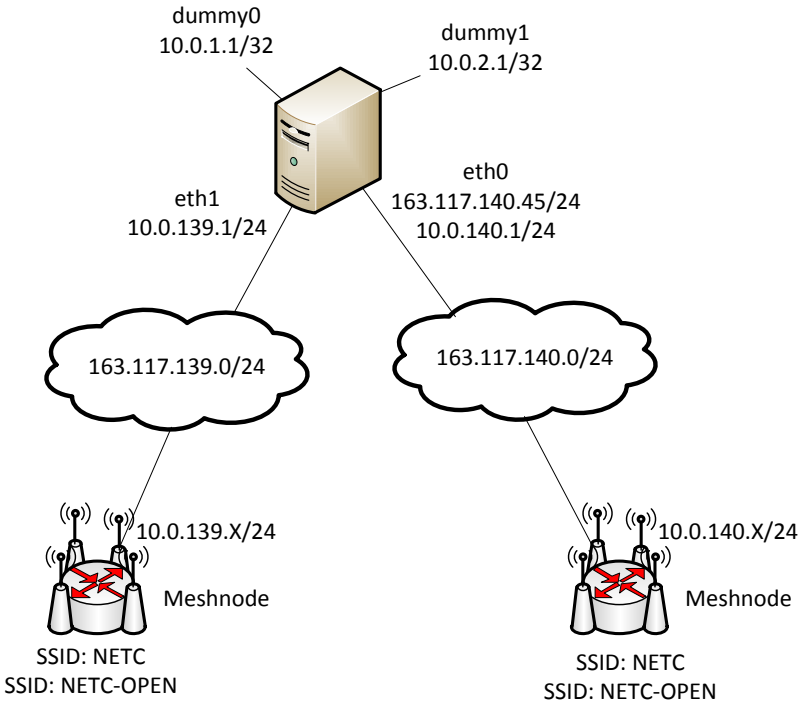

Figura 5.4: Estructura de la red

- meshnode02: 4.1.F11
- meshnode03: 4.0.F10
- meshnode04: 4.1.F15
- meshnode05: 4.1.A14
- meshnode06: 4.1.A03
- meshnode07: 4.1.C01
- meshnode08: 4.1.F04

En las figuras 5.5 y 5.6 se puede consultar la colocación de los nodos en los planos del edificio.

5.6.4. Conectividad inalámbrica entre los meshnodes

5.6.4.1. Introducción

Finalmente, después de tener los nodos colocados por el departamento, se procedió a colocar antenas para crear enlaces inalámbricos entre los nodos.

La solución creada con el OpenVPN para que el Chilli estuviera a un salto de los puntos de acceso, sigue siendo válida si se utilizara la red mesh para conectar con el servidor, en lugar de usar la red cableada. Para utilizar la red mesh, habría que modificar las rutas de los nodos.

Para las pruebas de movilidad que se explican posteriormente, no se ha utilizado la conectividad inalámbrica entre los nodos, sino que se han realizado a través de la red cableada.

5.6.4.2. Tipos de antenas

Para crear los enlaces se disponían de los siguientes tipos de antenas:

- Antenas omnidireccionales de 2.4GHz: se usan para los puntos de accesos al que se van a conectar a los usuarios. Todos los nodos que se han desplegado tienen una antena de este tipo conectada. Se utilizan estas ya que radian hacia todas las direcciones y tienen una mayor cobertura que las omnidireccionales de 5GHz.
- Antenas omnidireccionales de 5GHz: se han utilizado para comunicar nodos cercanos, ya que se obtiene igual ancho de banda que una sectorial de 5GHz y tienen menor coste económico.
- Antenas sectoriales de 5GHz: se han utilizado para comunicar nodos entre lo que hubiera bastante distancia entre ellos y no muchos obstáculos como paredes o armarios. Han sido las que más se han utilizado, ya que es una banda en la que no hay casi interferencias por lo que el ancho de banda es mayor que en 2.4GHz. Deben estar bien apuntadas entre ellas para conseguir el ancho de banda máximo.
- Antenas parche de 2.4GHz: sólo se ha utilizado en un caso, ya que era necesario tener un gran alcance para conectar esos dos nodos. Con esta antena se consigue conectar los nodos pero el ancho de banda medido es menor porque es una banda con muchas interferencias.
- Antenas parche de 5GHz: no se han utilizado ya que están reservadas para casos críticos.

5.6.4.3. Enlaces en el despliegue

En las figuras 5.5 y 5.6 se muestran la colocación de los nodos, los enlaces con sus respectivas antenas en el plano del edificio.

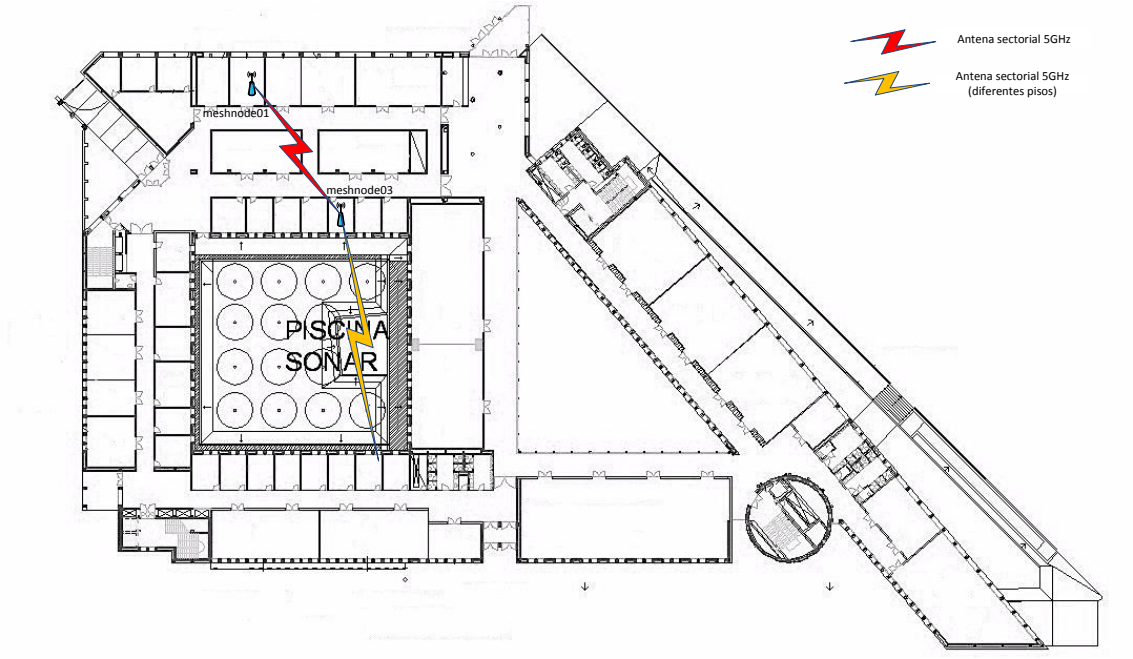

Figura 5.5: Plano planta baja edificio Torres Quevedo

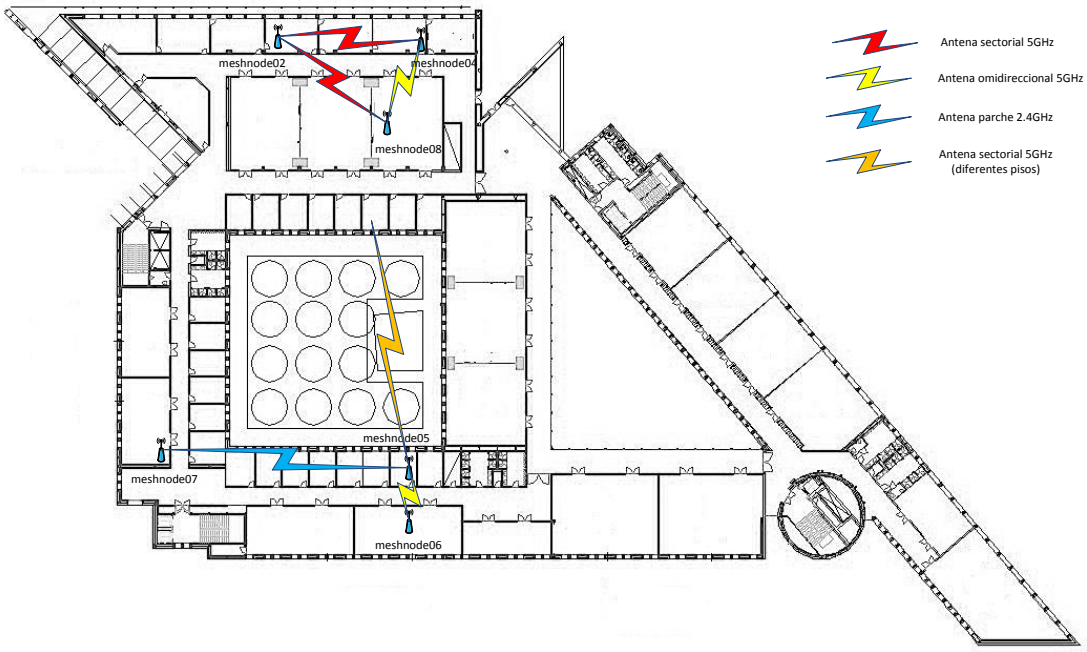

Figura 5.6: Plano primera planta edificio Torres Quevedo

5.7. Conclusiones

La red desplegada es una red de acceso Wi-Fi cuyos usuarios podrán navegar en Internet después de haber sido autenticados en la red.

La red desplegada es una red *mesh*, en la que los nodos se pueden comunicar entre ellos no sólo por el cable, sino también por los enlaces creados con las antenas. Además de

comunicarse con los nodos cercanos con las antenas, también se pueden comunicar con otros nodos que no estén al alcance a través de nodos intermedios, ya que la red es multisalto.

La red desplegada tiene una mayor flexibilidad de la que se va a hacer uso, ya que no se utilizan los enlaces inalámbricos. Estos enlaces están disponibles para futuros proyectos.

Capítulo 6

Estudio de soluciones de movilidad

6.1. Introducción

El objetivo de este Trabajo Fin de Grado es estudiar el rendimiento de las soluciones de movilidad en una red real. Se ha probado una solución a nivel de enlace (movilidad en una red Wi-Fi), y dos soluciones a nivel IP, MIPv6 y PMIPv6, cuyo funcionamiento se explicó en los capítulos 3 y 4 respectivamente.

El estudio mide el rendimiento de las soluciones midiendo el tiempo total del traspaso desde que el nodo móvil recibe el último paquete hasta que recibe el primer paquete después de ejecutar los mecanismos de movilidad. También se analizan los componentes intermedios que afectan a la movilidad.

Todas las pruebas se han hecho con IPv6, ya que los protocolos MIPv6 y PMIPv6 están diseñados para ello. A nivel de enlace se ha anunciado un prefijo IPv6 para que la comparación fuera justa.

Además, en este estudio, se va a observar como afecta a la movilidad el hecho de introducir cierto retardo entre los elementos intermedios que participan.

6.2. Arquitectura y equipos utilizados

Para las pruebas se ha utilizado el diagrama de red que se muestra en la figura 6.1.

A continuación se explica funcionalidad de cada uno de ellos y qué equipos se han utilizado:

- **Servidor**: ejecutará el portal cautivo para la solución de nivel de enlace y realizará las funciones de agente local en MIPv6 y de LMA en PMIPv6. Además en una de sus interfaces *dummy* estará localizado el nodo corresponsal, que generará tráfico hacia nodo móvil.

  Se ha utilizado el equipo *escorpion*. Para ejecutar las implementaciones de MIPv6 y PMIPv6 se ha compilado un kernel personalizado como se explica en el apéndice D.

- **Nodos inalámbricos**: en una de sus interfaces se crean los puntos de acceso para que el nodo móvil se conecte a él. En nivel de enlace realizan las funciones de puntos

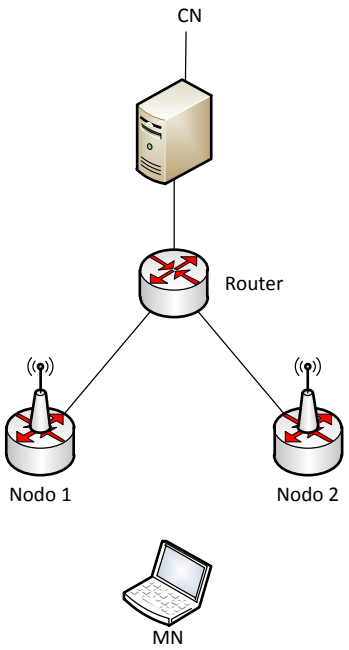

Figura 6.1: Arquitectura de la red

de acceso, en MIPv6, realizan las acciones de un router de acceso (Access Router, AR); y en PMIPv6 actúan como MAG.

Se utilizaron dos de los nodos desplegados: meshnode05 y meshnode06, situados en los despachos 4.1A.14 y 4.1A.03, respectivamente. Para actuar como MAG, fue necesario compilar un kernel personalizado con las opciones que se explican en el anexo D.

- **Nodo móvil**: este equipo se conectará a los nodos inalámbricos. Sólo necesita de una configuración especial en MIPv6, ya que necesitará ejecutar los mecanismos de movilidad del protocolo. En los otros dos casos, al ser movilidad gestionada por la red, no necesita de software adicional.

  Se ha usado el equipo *gamusino* situado en el aula 4.1A03. Este equipo dispone de una tarjeta inalámbrica Atheros AR5001X+, con el driver ath5k. Al igual que con el servidor y los nodos inalámbricos, hubo que compilar un kernel para MIPv6 (Anexo D).

- **Router**: se introdujo para que simulara un retardo entre los nodos inalámbricos y el servidor. Con este elemento se pretende estudiar el impacto que tiene el retardo sobre los mecanismos de movilidad.

  Se utilizó el PC *capullo* disponible el aula 4.1A03 y que disponía dos tarjetas Ethernet para poder conectase tanto a la red 163.117.140.0/24 como a la red 163.117.139.0/24.

6.3. Simulación de retardo

Se introdujo un cierto retardo para simular un área más amplia entre los nodos inalámbricos y el servidor. Con ello se puede ver el impacto que tiene el retardo sobre los protocolos.

Como se ha comentado, para simular el retardo se introdujo un router, que atravesaría

todo el tráfico dirigido al nodo corresponsal o al nodo móvil. Los nodos estaban situados en las redes 163.117.140.0/24 y 163.117.139.0/24, por lo que fue necesario conectar el router a ambas redes, al igual que el servidor. Con esto se tiene el router conectados a las redes. En la figura 6.2 se observa como están interconectados todos lo elementos físicamente.

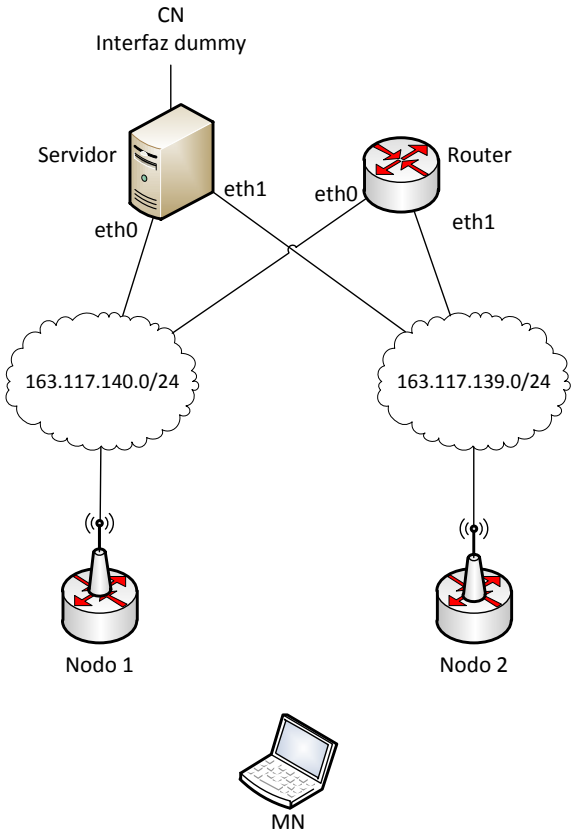

Figura 6.2: Estructura de la red física al simular retardo

Después, configurando rutas estáticas, se consiguió que todo el tráfico atravesara este router. Fue necesario configurar rutas IPv6, para MIPv6 y PMIPv6, y rutas IPv4 para que el túnel atravesara el router en el caso de nivel de enlace. El router, para enviar paquetes hacia el servidor, siempre utiliza la red 163.117.140.0/24 aunque podrían haberse comunicado con él por la red 163.117.139.0/24. Con ello se conseguía tener una estructura lógica como la que se observa en la figura 6.3, en la que el tráfico atraviesa el túnel.

Al reenviar el tráfico por la red en la que estaba conectado el servidor y uno de los nodos, el router recibía un paquete por la misma interfaz por la que luego lo enviaba. En consecuencia enviaba mensajes de redirección (ICMP Redirect) para informar de que existía una ruta directa. Tanto el nodo afectado como el servidor lo aprendían por lo que enviaban tráfico directamente sin tener el efecto deseado. Por lo que hubo que configurar que el router no enviara los mensaje de redirección y que el servidor y el nodo afectado no aceptaran el mensaje tanto para IPv4 como para IPv6.

Finalmente, se utilizó la herramienta netem para introducir retardos. En las pruebas, siempre se introdujo un retardo fijo sin ninguna variación estadística. Para introducir el retardo hay que indicarle cuanto retardo se le quiere introducir y la interfaz en la que lo debe aplicar.

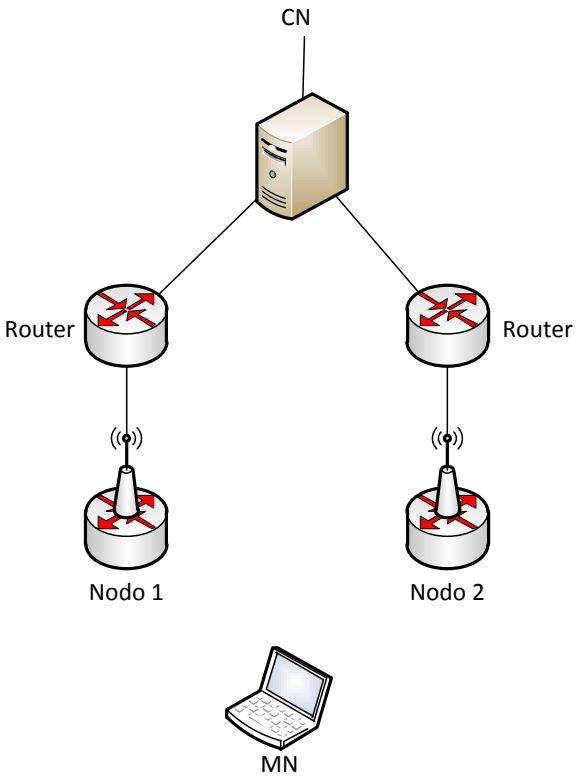

Figura 6.3: Estructura de la red lógica al simular retardo

6.4. Pruebas

El objetivo de las pruebas es medir el tiempo total en el que el nodo móvil tarda en hacer el traspaso desde un punto de acceso al otro. Para medir el tiempo, se genera tráfico desde el nodo corresponsal al nodo móvil, midiendo desde el último paquete recibido antes de iniciar el traspaso hasta que se recibe el primero después de este.

Para generar tráfico se utiliza el comando `ping6`. Se generaba un paquete cada 10 ms, ya que fue el menor tiempo entre pings que el servidor enviaba correctamente. Con intervalos de pings más pequeños se observó que no era preciso.

Además se utilizó la herramienta `tshark` para capturar el tráfico en los nodos inalámbricos, en el nodo móvil y en el servidor, para posteriormente, analizar las tramas y medir el tiempo total y los tiempos individuales. `Tshark` es un analizador de protocolos igual que Wireshark, salvo que no tiene una interfaz gráfica y la salida se obtiene en un fichero de texto.

Para que el nodo móvil cambiara de punto de acceso se ha ejecutado el comando `iwconfig` incluyendo el SSID al que se deseaba cambiar. En la movilidad a nivel de enlace los dos puntos de acceso mostraban el mismo SSID, por lo que, además, era necesario especificar la dirección MAC de cada punto en el comando. En los casos de MIPv6 y PMIPv6, las dos redes tenían SSIDs diferentes.

En las pruebas siempre se busca que el tiempo fuera el menor posible, por lo que se eliminó el escaneo de redes en el nodo móvil. Para ello, antes de cambiar de punto de acceso, se ejecutaba el comando `iwlist wlan0 scan`, que hacía un escaneo activo en todos los canales. Al llegar el momento de cambiar de punto de acceso, el nodo móvil tenía

guardado en memoria los SSIDs y el canal de cada uno, por lo que no volvía a ejecutar ningún escaneo y se conseguía eliminar este tiempo.

Finalmente, se escribieron varios *scripts* con los que se automatizaban las pruebas y al final, se obtenía un fichero con todos los tiempos medidos en cada una de las pruebas.

6.4.1. Nivel de enlace

El software utilizado a nivel de enlace ha sido el portal cautivo junto con los túneles, que se ha explicado en el capítulo 5 y con más detalle en los anexos B y C. En esta solución todos los puntos de acceso presentaban el mismo SSID.

El portal cautivo está desarrollado para que el nodo móvil obtuviera un dirección IPv4. Para que la comparación con el resto de protocolos fuera justa, se decidió anunciar un prefijo IPv6 por el túnel. Con ello el nodo móvil configuraría un dirección IPv6 en la interfaz inalámbrica con los mecanismos de autoconfiguración.

Al empezar las pruebas, se observó que, al conectarse un nuevo cliente a un punto de acceso, el programa OpenVPN enviaba un paquete de nivel de enlace XID (eXchange Identification) en el que se informaba a toda la subred en qué punto de acceso estaba conectado el nodo. Con este paquete se consigue que el servidor, es decir, el programa Chilli aprendiera donde se encontraba el nodo móvil y poder reenviar el tráfico al punto de acceso correspondiente. Este mecanismo se ha explicado en el capítulo 3.

Para analizar los componentes en los que se dividía el traspaso se definió el diagrama de secuencia representado en la figura 6.4 donde se definen los siguientes tiempos:

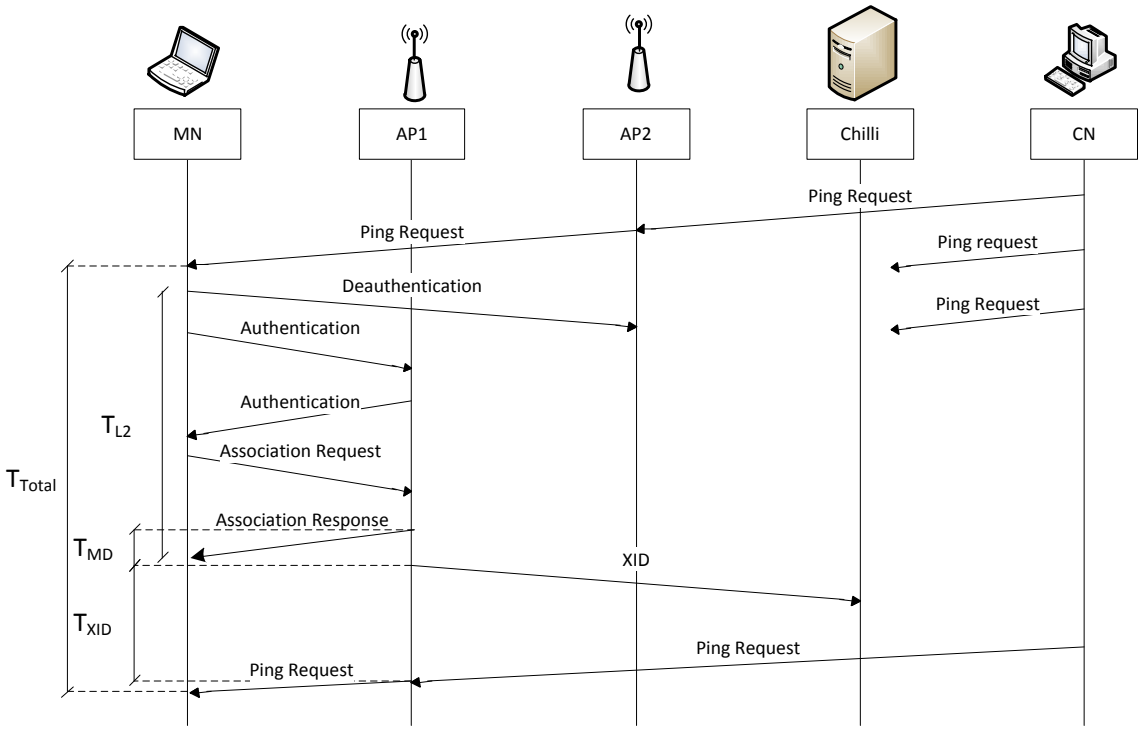

Figura 6.4: Diagrama de señalización del protocolo de movilidad a nivel de enlace

- Tiempo de traspaso a nivel dos (Layer 2 handover, T_{L2}): es el tiempo en el que se

produce la señalización a nivel de enlace por la que el cliente cambia de punto de acceso. Para ello, se desautentica del antiguo punto de acceso y se autentica y se asocia con el nuevo.

- Detección de movimiento (Movement detection, T_{MD}): es el tiempo que tarda el punto de acceso en conocer que un cliente se ha conectado. Se define este tiempo desde que el punto de acceso envía el mensaje de asociación a la estación (Association Response) hasta que envía el mensaje a nivel de enlace al resto de la subred.

- Tiempo XID (T_{XID}): se mide como el tiempo desde que el punto de acceso envía el mensaje XID hasta que recibe el primer ping. En este tiempo el servidor aprende donde se encuentra el nodo móvil y cambia su tabla de reenvío. Se ha medido de esta forma ya que el mensaje XID no tiene un mensaje de confirmación, por lo que el único mensaje por el que se puede medir que el servidor conoce la nueva localización del nodo móvil, es por el primer ping request.

En las figuras 6.5 y 6.6 se puede ver esta señalización capturada con Wireshark en el nodo móvil y en el punto de acceso. Se observa como envía el mensaje XID, por el que el servidor aprende donde se encuentra el nodo móvil y envía los paquetes al punto de acceso correspondiente.

No.	Time	Source	Destination	Protocol	Info
39	15.302804000	fe80::685f:b6ff:fe20:5c35	ff02::1	ICMPv6	Router Advertisement from 6a:5f:b6:20:5c:35
40	16.139539000	Netgear_61:16:7c	WistronN_00:49:77	802.11	Authentication, SN=3306, FN=0, Flags=........C
42	16.140733000	WistronN_00:49:77	Netgear_61:16:7c	802.11	Authentication, SN=3090, FN=0, Flags=........
43	16.141399000	Netgear_61:16:7c	WistronN_00:49:77	802.11	Association Request, SN=3307, FN=0, Flags=........C, S
45	16.142681000	WistronN_00:49:77	Netgear_61:16:7c	802.11	Association Response, SN=3091, FN=0, Flags=........
46	16.147350000	Netgear_61:16:7c	Broadcast	XID	Basic Format; Type 1 LLC (Class I LLC); Window Size 0
47	16.167283000	2001:720:410:1017::1	2001:720:410:1017:20f:b5ff:fe61:167c	ICMPv6	Echo (ping) request id=0x264d, seq=8108
49	16.168976000	2001:720:410:1017::1	2001:720:410:1017:20f:b5ff:fe61:167c	ICMPv6	Echo (ping) request id=0x264d, seq=8108
50	16.170301000	2001:720:410:1017:20f:b5ff:fe61:167c	2001:720:410:1017::1	ICMPv6	Echo (ping) reply id=0x264d, seq=8108
51	16.170377000	2001:720:410:1017:20f:b5ff:fe61:167c	2001:720:410:1017::1	ICMPv6	Echo (ping) reply id=0x264d, seq=8108
52	16.187717000	2001:720:410:1017::1	2001:720:410:1017:20f:b5ff:fe61:167c	ICMPv6	Echo (ping) request id=0x264d, seq=8109
54	16.190930000	2001:720:410:1017::1	2001:720:410:1017:20f:b5ff:fe61:167c	ICMPv6	Echo (ping) request id=0x264d, seq=8109
55	16.192319000	2001:720:410:1017:20f:b5ff:fe61:167c	2001:720:410:1017::1	ICMPv6	Echo (ping) reply id=0x264d, seq=8109

Figura 6.5: Señalización capturada en el Wireshark en el nuevo punto de acceso

No.	Time	Source	Destination	Protocol	Info
6642	21.180827	2001:720:410:1017::1	2001:720:410:1017:20f:b5ff:	ICMPv6	Echo (ping) request id=0x264d, seq=8104
6644	21.181008	2001:720:410:1017:20f:b5ff:	2001:720:410:1017::1	ICMPv6	Echo (ping) reply id=0x264d, seq=8104
6647	21.204875	2001:720:410:1017::1	2001:720:410:1017:20f:b5ff:	ICMPv6	Echo (ping) request id=0x264d, seq=8105
6649	21.205059	2001:720:410:1017:20f:b5ff:	2001:720:410:1017::1	ICMPv6	Echo (ping) reply id=0x264d, seq=8105
6653	21.215432	Netgear_61:16:7c	WistronN_02:01:9f	802.11	Deauthentication, SN=1645, FN=0, Flags=........
6657	21.250939	Netgear_61:16:7c	WistronN_00:49:77	802.11	Authentication, SN=1646, FN=0, Flags=........
6658	21.251501	WistronN_00:49:77	Netgear_61:16:7c	802.11	Authentication, SN=3404, FN=0, Flags=........C
6660	21.252831	Netgear_61:16:7c	WistronN_00:49:77	802.11	Association Request, SN=1647, FN=0, Flags=........, SSID=NETC-OPEN
6661	21.253449	WistronN_00:49:77	Netgear_61:16:7c	802.11	Association Response, SN=3405, FN=0, Flags=........C
6662	21.279741	2001:720:410:1017::1	2001:720:410:1017:20f:b5ff:	ICMPv6	Echo (ping) request id=0x264d, seq=8108
6664	21.281737	2001:720:410:1017:20f:b5ff:	2001:720:410:1017::1	ICMPv6	Echo (ping) reply id=0x264d, seq=8108
6666	21.301705	2001:720:410:1017::1	2001:720:410:1017:20f:b5ff:	ICMPv6	Echo (ping) request id=0x264d, seq=8109
6668	21.303756	2001:720:410:1017:20f:b5ff:	2001:720:410:1017::1	ICMPv6	Echo (ping) reply id=0x264d, seq=8109

Figura 6.6: Señalización capturada en el Wireshark en nodo móvil

6.4.2. MIPv6

Para las pruebas del protocolo MIPv6 se utilizó la implementación UMIP [umi] con licencia de software libre. Como se ha comentado, este software necesitaba un kernel con ciertas opciones, por lo que fue necesario compilar un nuevo kernel para el servidor y para el equipo que actúa como nodo móvil. En el anexo D se puede consultar los pasos para compilar el kernel y los archivos de configuración del software. Este kernel sólo era necesario instalarlo en los equipos que actúan como agente local y como nodo móvil.

Con este software puede usarse los dos modos de funcionamiento de MIPv6: encaminar los paquetes a través del agente local o utilizar el mecanismo de optimización de rutas. En nuestras pruebas se utilizó el reenvío a través del agente local.

También se configuró para que la detección de movimiento fuera lo más rápida que permite el estándar sin utilizar interacciones de nivel dos. Para ello se configuraron los routers de acceso para que enviasen Routers Advertisement (RA) con el mínimo tiempo estandarizado, entre 30 ms y 70 ms.

Además se ha utilizado la configuración más "agresiva" (*eager*) del programa, en la que al recibir un nuevo prefijo de un router configura una dirección IP con ese prefijo y lo anuncia al agente local. Con lo que no ejecuta ningún mecanismo de Neighbor Unreachability Detection (NUD) para verificar que realmente se ha cambiado de enlace o si sólo hay otro router en el enlace.

Asimismo el nodo móvil se configuró para que no ejecutara el mecanismo de detección de dirección duplicada (Duplicate Address Detection, DAD) de IPv6, ya que esto añade aproximadamente un segundo al tiempo total y no se hace una comparación justa con el resto de soluciones.

En la figura 6.7 se muestran los distintos tiempos de la señalización que se produce en MIPv6 y que forman el handover total.

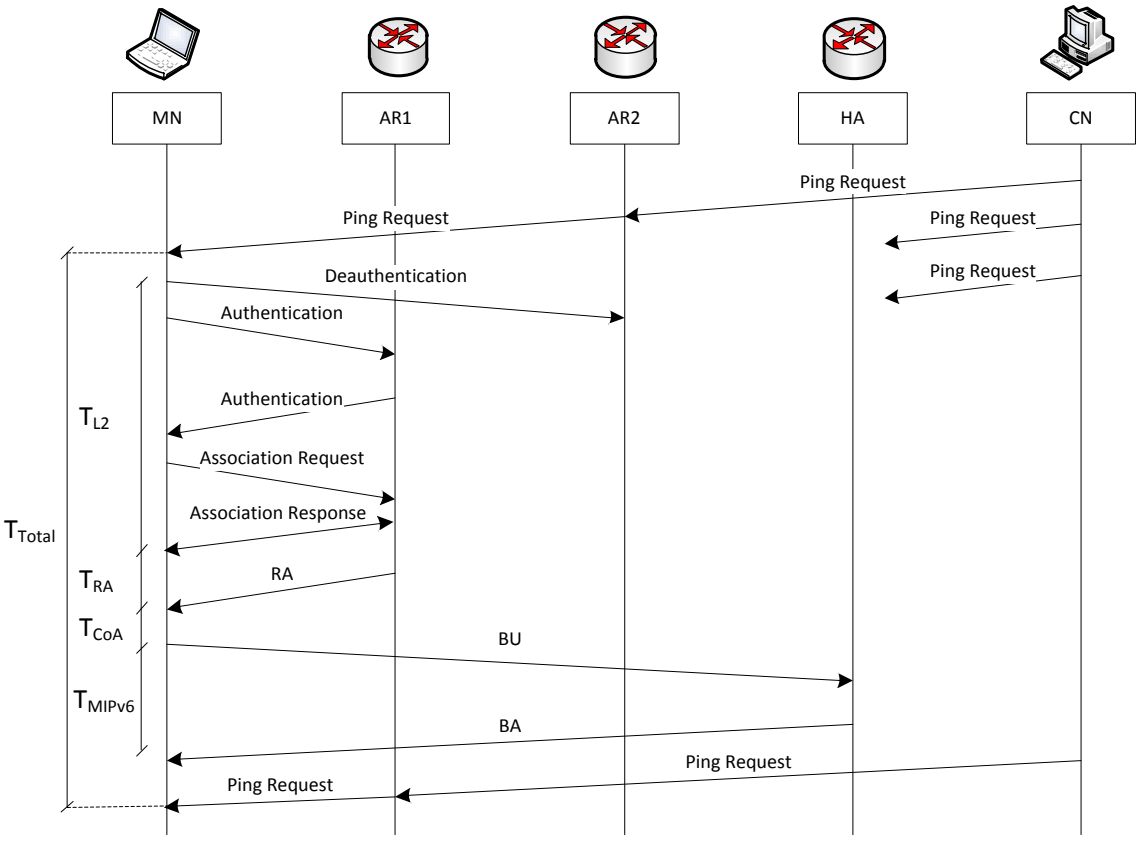

Figura 6.7: Diagrama de señalización del protocolo MIPv6

- Tiempo de traspaso a nivel dos (Layer 2 handover, T_{L2}): es el mismo tiempo que se ha descrito en la sección anterior.

- Tiempo hasta que recibe un Router Advertisement (T_{RA}): es el tiempo desde que se ha completado la asociación con el punto de acceso hasta que recibe el primer Router Advertisement (RA). Como se ha comentado este tiempo se ha configurado según los valores mínimo establecidos en [PJA11], entre 30 ms y 70 ms.
- Tiempo de configuración de la dirección CoA (T_{CoA}): es el tiempo que tarda el equipo en configurar la dirección IP en la red visitada CoA. Se mide desde que se recibe el primer RA hasta que envía el Binding Update (BU) al agente local para notificar su nueva IP.
- Tiempo de señalización MIPv6 (T_{MIPv6}): es el tiempo que transcurre desde que el nodo móvil envía un Binding Update (BU) hasta que recibe la confirmación del agente local con el mensaje Binding Acknowledgement (BA). En este tiempo el agente local configura el túnel y las rutas necesarias para alcanzar al nodo móvil.

En la figura 6.8 se observa los mensajes intercambiados en MIPv6 capturados en Wireshark en el nodo móvil. Además se puede ver como en los mensajes de ping request y ping response se produce la encapsulación con dos cabeceras IPv6.

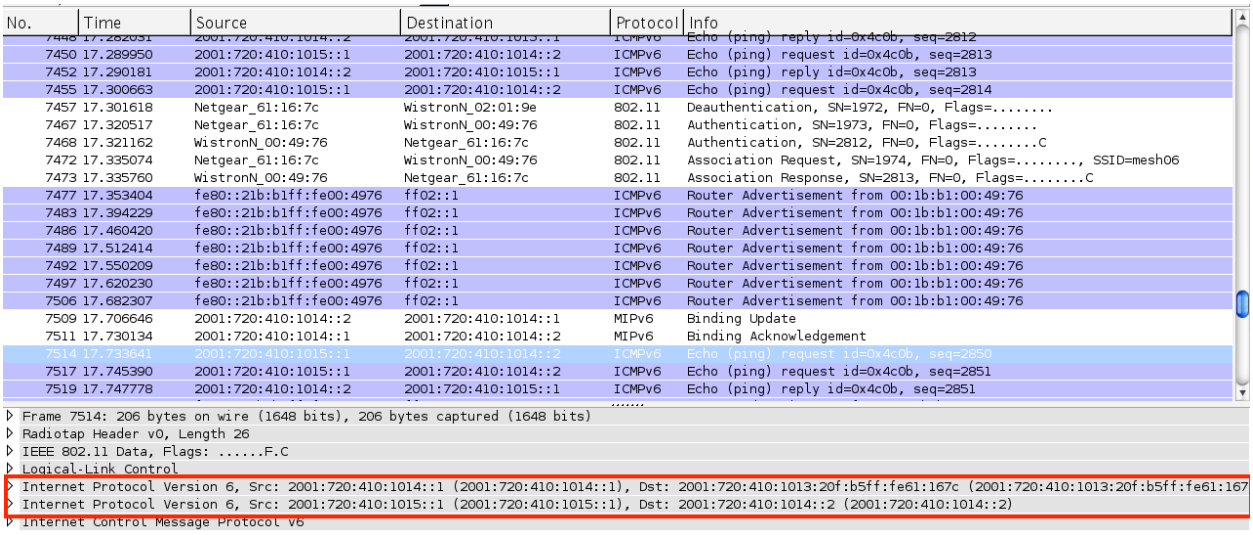

Figura 6.8: Señalización capturada durante el traspaso en MIPv6 en el nodo móvil

6.4.3. PMIPv6

Para las pruebas de PMIPv6 se utilizó el software desarrollado por EURECOM, que se basa en la implementación UMIP, y que se denomina OpenAirInterface Proxy Mobile IPv6 (OAI PMIPv6) [pmi]. Al igual que PMIPv6, tiene licencia de software libre. Al igual que en MIPv6, en los MAGs hubo que instalar un kernel con las opciones de necesarias para ejecutar el programa.

Esta implementación no cumple con el estándar definido en la RFC [LJW11], por las siguientes razones:

- En los mensaje PBA, dentro de Mobility Options los campos Handoff Indicator option y Access Technology Type option no están presentes. Según el estándar son obligatorios.

- En el mensaje PBA, el bit P, de registro Proxy, no está a 1.

  Estos errores se pueden ver en la figura 6.9. Los bits que aparecen como "[IE data no dissected]" corresponden al campo anterior, Mobile Node Link-layer Identifier option.

Figura 6.9: Errores en los mensajes PBA

Al no cumplir con el estándar se decidió probar una versión desarrollada por UMIP. Pero esta versión no se pudo utilizar ya los MAGs daban error al recibir un PBA. Por lo que finalmente se decidió usar la implementación OAI PMIPv6, ya que cumple la funcionalidad pero no con el estándar.

La señalización y los tiempos que se producen en PMIPv6 son los que se describen en la figura 6.10.

- Tiempo de traspaso a nivel dos (Layer 2 handover, T_{L2}): es el mismo tiempo que se ha descrito en la solución a nivel de enlace y en MIPv6.
- Envío de Router Solicitation (T_{RS}): es el tiempo desde que se ha completado el traspaso a nivel de enlace hasta que el nodo móvil envía un RS necesario para el MAG detecte su presencia. Para que este tiempo fue el menor posible se enviaban un mensaje RS cada 1ms.
- Detección de movimiento (Movement Detection, T_{MD}): en este tiempo, el MAG detecta que un nuevo nodo móvil se ha conectado. Es el tiempo que transcurre desde que el MAG recibe el RS del nodo móvil hasta que envía el PBU para informar al LMA.
- Señalización PMIPv6 (T_{PMIPv6}): es el tiempo desde que el MAG envía el PBU hasta que recibe el PBA del LMA. En este tiempo el LMA configura el túnel con el MAG y todas las rutas necesarias.
- Envío Router Advertisement (T_{RA}): es el tiempo de procesamiento del PBA en el MAG, en el configura el túnel y las rutas necesarias. Se mide desde que recibe el PBA del LMA hasta envía el RA al nodo móvil.

Los tiempos de propagación (T_{Prop}) son ignorados, ya que se consideran mucho menor que el tiempo total del traspaso.

En las figuras 6.11 y 6.12 se puede ver la señalización capturada con el programa Wireshark en el nuevo MAG y en el nodo móvil. En la figura 6.11 se puede ver el túnel entre el MAG y el LMA en los paquetes destinados al nodo móvil.

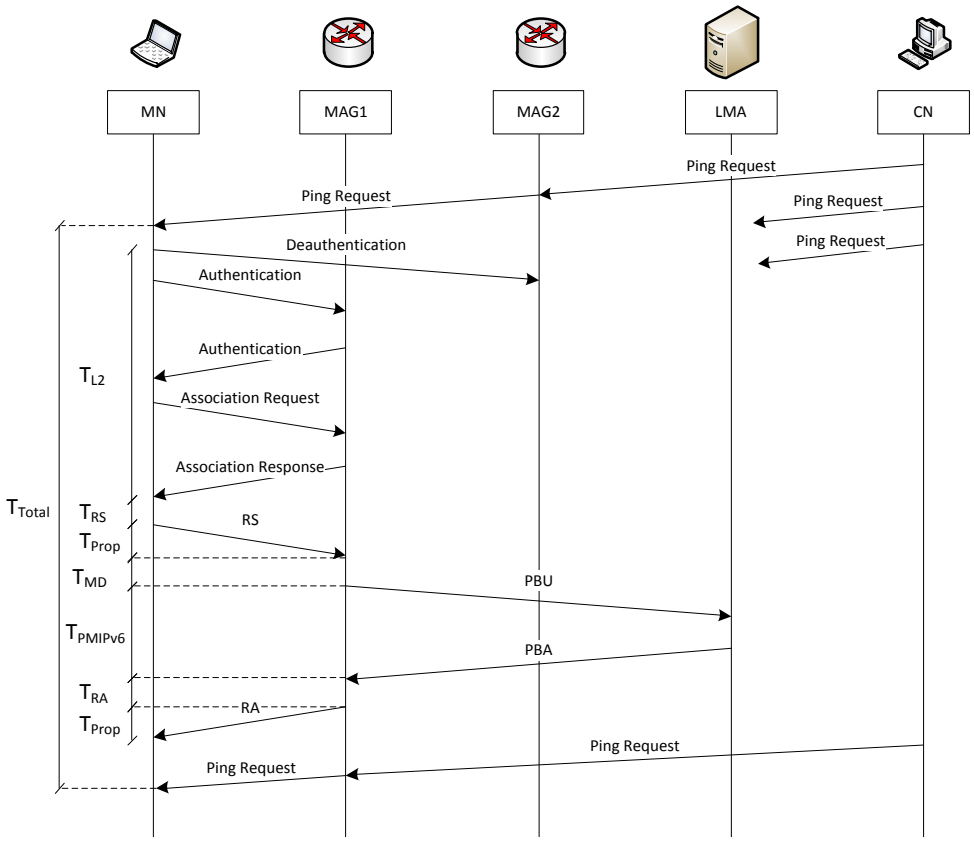

Figura 6.10: Diagrama de señalización del protocolo PMIPv6

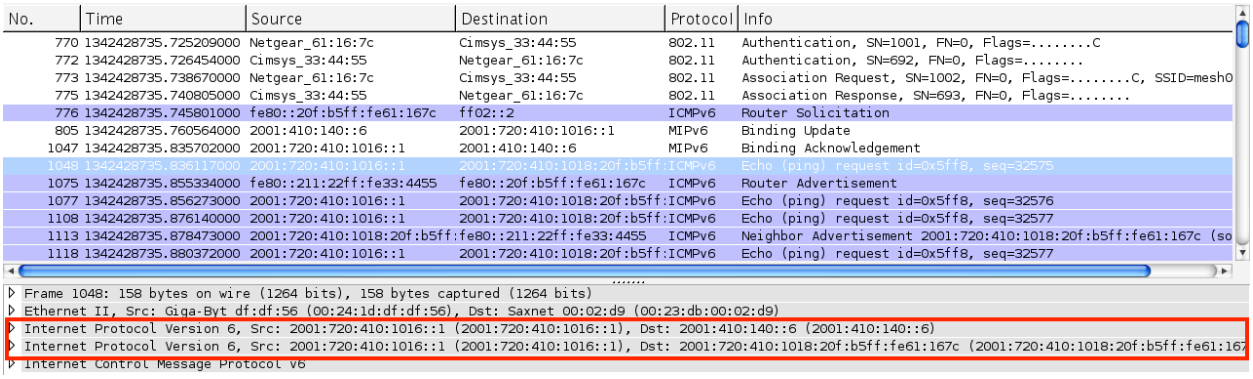

Figura 6.11: Señalización capturada en el nuevo MAG

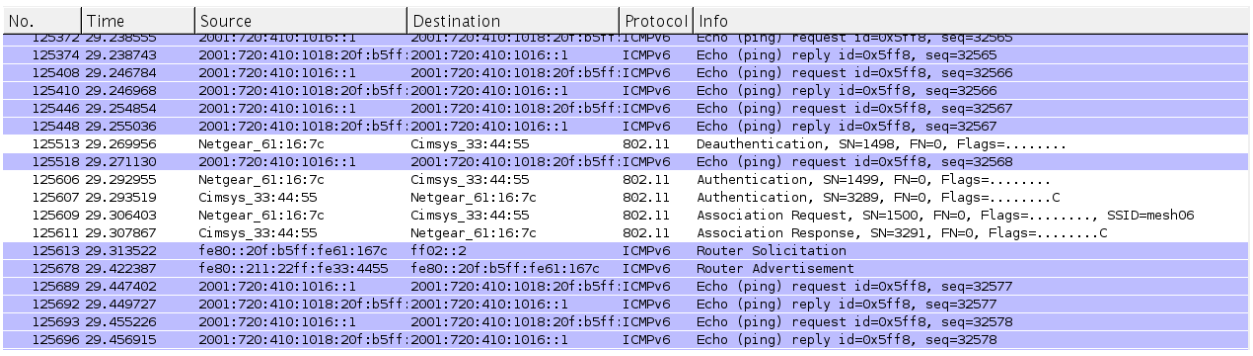

Figura 6.12: Señalización capturada en el nodo móvil

6.5. Resultados

6.5.1. Introducción

Los resultados obtenidos y representados en las gráficas se han obtenido con los datos de 100 ejecuciones por cada retardo añadido. El retardo se ha añadido desde 0 ms a 260 ms en saltos de 20ms.

Para dibujar las gráficas se ha usado el programa `gnuplot`.

En algunas figuras se representa el diagrama *box-plot* o *box-and-whisker* en el que el límite inferior del rectángulo es el primer cuartil, el límite superior, el tercer cuartil; y la banda en el medio del rectángulo, es la mediana. Los límites de los *whiskers* que extienden desde cada límite del rectángulo pueden representar diferentes valores. Por defecto en `gnuplot` y en nuestro caso, se representa el límite del rectángulo hasta un rango igual a 1.5 veces el rango intercuartil (IQR), el cual es igual a la resta del tercer y el primer cuartil. Y las cruces por debajo o por encima de los *whiskers*, representan valores atípicos (*outliers*).

6.5.2. Resumen

En la figura 6.13 se representa la función de distribución acumulada (Cumulative Distribution Function, CDF) de cada solución, en la que se puede ver el tiempo total de cada una de las soluciones sin ningún tipo de retardo.

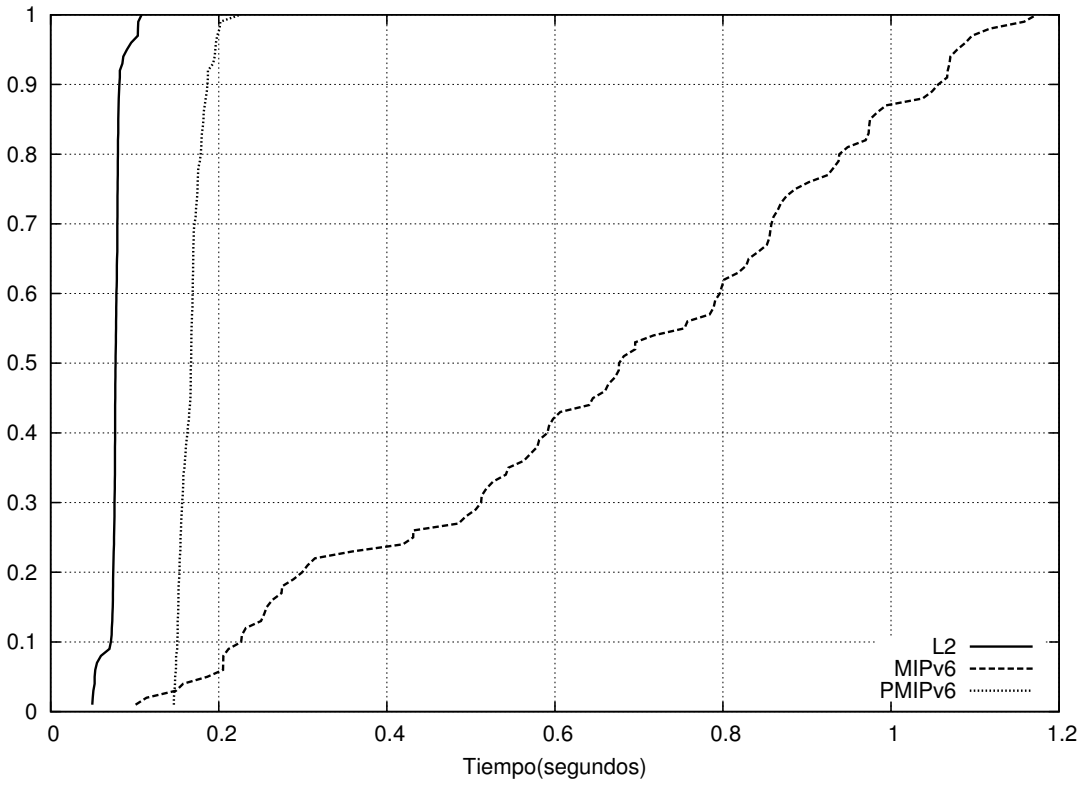

Figura 6.13: CDF: Resumen de los resultados

Se observa como la mejor solución es la movilidad a nivel de enlace (L2), ya que es en

la que siempre se obtiene un mejor tiempo de traspaso. El peor caso es MIPv6, ya que su tiempo es muy variable teniendo en el mejor de los casos un retardo de 100ms y en el peor, casi 1.2 segundos. El tiempo de traspaso en PMIPv6 es algo más lento que en el caso de nivel de enlace, pero mucho menor que el tiempo de traspaso de MIPv6.

A las siguientes secciones se van a mostrar las diferentes componentes de cada una de las soluciones.

6.5.3. Nivel de enlace

En la figura 6.14 se presenta la gráfica *boxplot* con los resultados individuales acumulados y el tiempo total obtenidos en movilidad a nivel de enlace sin ningún retardo añadido.

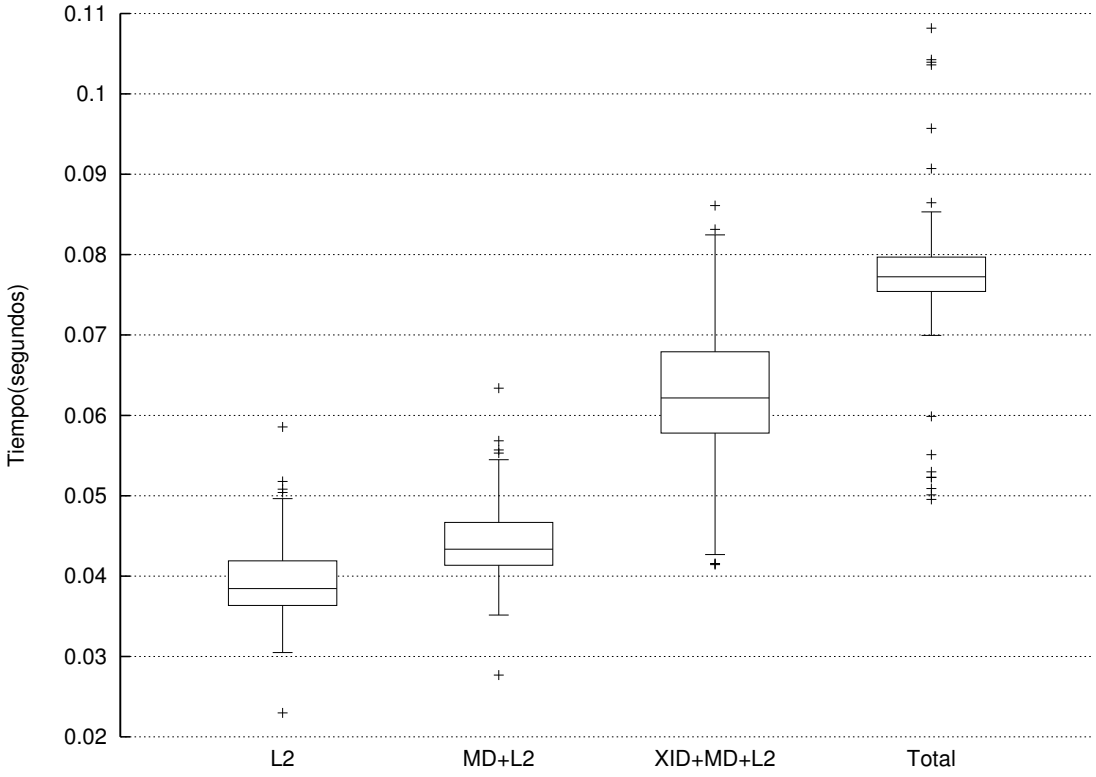

Figura 6.14: Resultados obtenidos en movilidad a nivel de enlace

Se observa que en el peor caso se tiene un traspaso total de casi unos 110 ms y en el mejor de los casos, 50 ms, encontrándose la mayoría de resultados entre 70 ms y 85 ms. Esta variabilidad depende de los pings que se envían cada 10 ms, según el momento en el que los envíe el nodo corresponsal.

Se observa que el tiempo que más afecta a la movilidad es el tiempo de traspaso a nivel dos del nodo móvil, ya que es la mitad del tiempo total del traspaso.

Al añadir el retardo, el tiempo al que afecta es el T_{XID}, haciendo aumentar el tiempo total del traspaso proporcionalmente al retardo añadido.

6.5.4. MIPv6

En la figura 6.15 se presentan los resultados de las pruebas de MIPv6 sin ningún retardo añadido. En esta gráfica se representa el diagrama *boxplot* de los tiempos individuales acumulados.

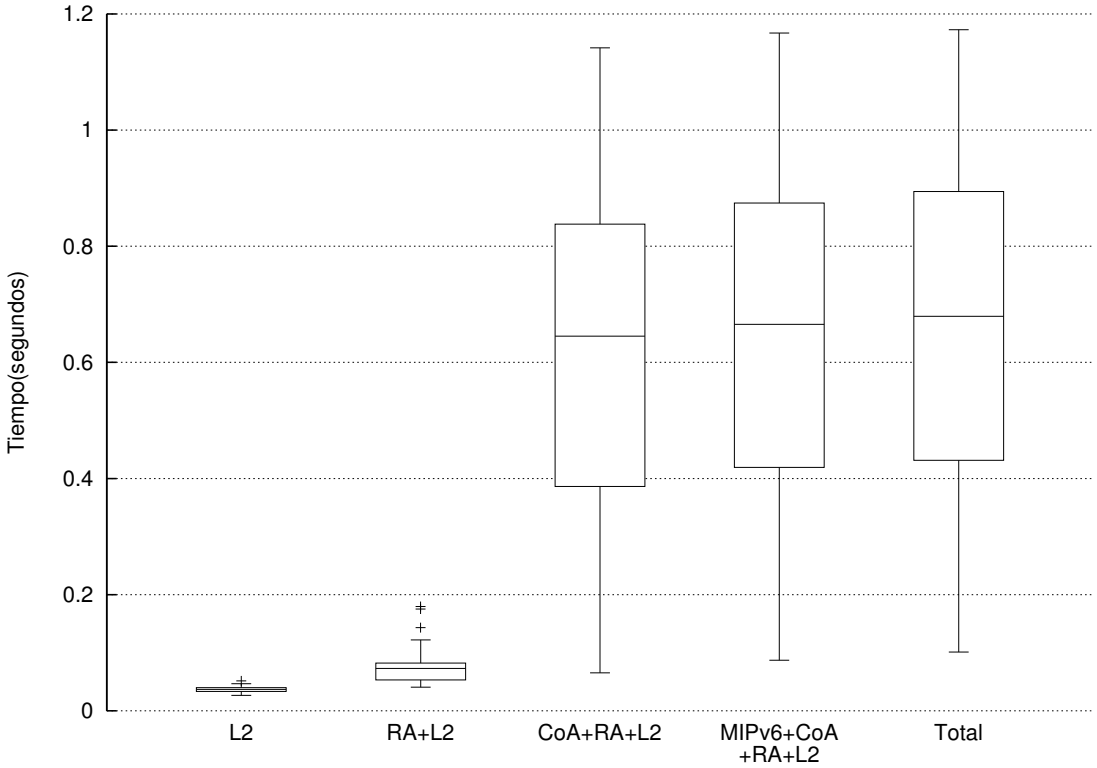

Figura 6.15: Resultado obtenidos en MIPv6

Se observa que el tiempo total de traspaso es muy variable donde el mínimo son unos 100ms y el máximo 1.15 segundos. Esta variabilidad se debe al tiempo de configuración de la dirección CoA, que varía desde 50ms hasta el segundo, siendo este tiempo el que más afecta al tiempo total del traspaso.

Como se ha comentado el tiempo de configuración de la dirección CoA se mide desde que el nodo móvil recibe un RA hasta que envía el BU al agente local. Este tiempo depende de la implementación del protocolo por lo que estudió el código fuente para reducir el tiempo o, por lo menos, no hacerlo tan variable. Se miró el código durante unas semanas y se preguntó a los desarrolladores sin conseguir solucionarlo. Finalmente se decidió dejarlo así ya que se salía de los objetivos del Trabajo Fin de Grado. En [XCZ+07], se puede observar que también tienen el mismo problema al medir el tiempo total.

Al añadir el retardo simulado, el tiempo total aumenta proporcionalmente al tiempo que se le añade. El aumento afecta al tiempo definido como T_{MIPv6}, que es la única señalización que atraviesa el enlace con retardo.

6.5.5. PMIPv6

En la figura 6.15 se pueden ver los resultados obtenidos con el protocolo PMIPv6. Al igual que en los dos casos anteriores, se representa el diagrama *box-plot* con los tiempos individuales acumulados.

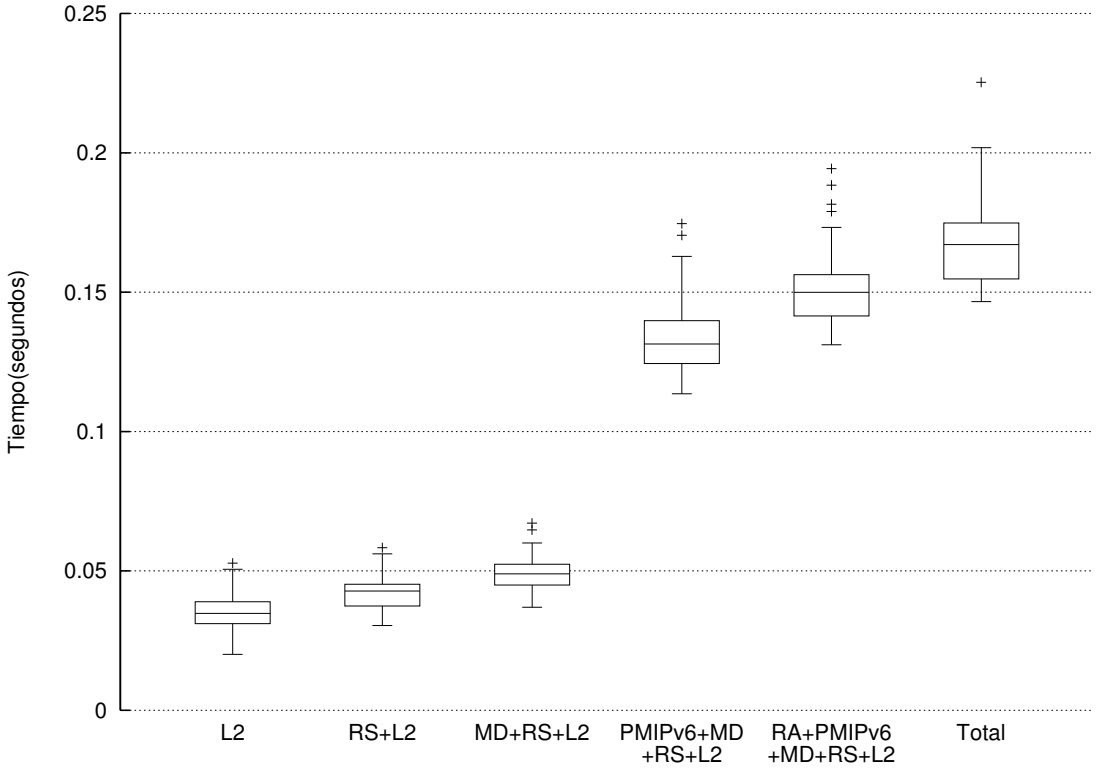

Figura 6.16: Resultado obtenidos en PMIPv6

En este caso el traspaso total varía desde unos 150ms hasta unos 200ms. Además, como se puede ver, el tiempo que más afecta al traspaso total es el tiempo T_{PMIPv6}, que mide desde que el MAG envía el PBU hasta que recibe el PBA. En este tiempo el LMA configura el túnel y las rutas necesarias.

Este tiempo se podría reducir, ya que observando los *logs* del programa el túnel que tenía creado con el anterior MAG lo elimina y crea uno nuevo con el nuevo MAG. En lugar de eso se podría modificar el túnel, lo que haría reducir el tiempo total de traspaso.

Al igual que en el caso MIPv6, el tiempo donde afecta el retardo introducido es en la señalización de movilidad, aumentando el tiempo total del *handover* proporcionalmente al retardo añadido.

6.5.6. Comparativa

En esta última sección se van a comparar los resultados de las soluciones estudiadas y se comprueba en qué componente afecta el retardo añadido.

En la figura 6.17 se representa un histograma con la media de las componentes de cada solución con diferentes retardos. Los retardos representan t1, t2 y t3, son 0 ms, 140 ms y

260 ms, respectivamente.

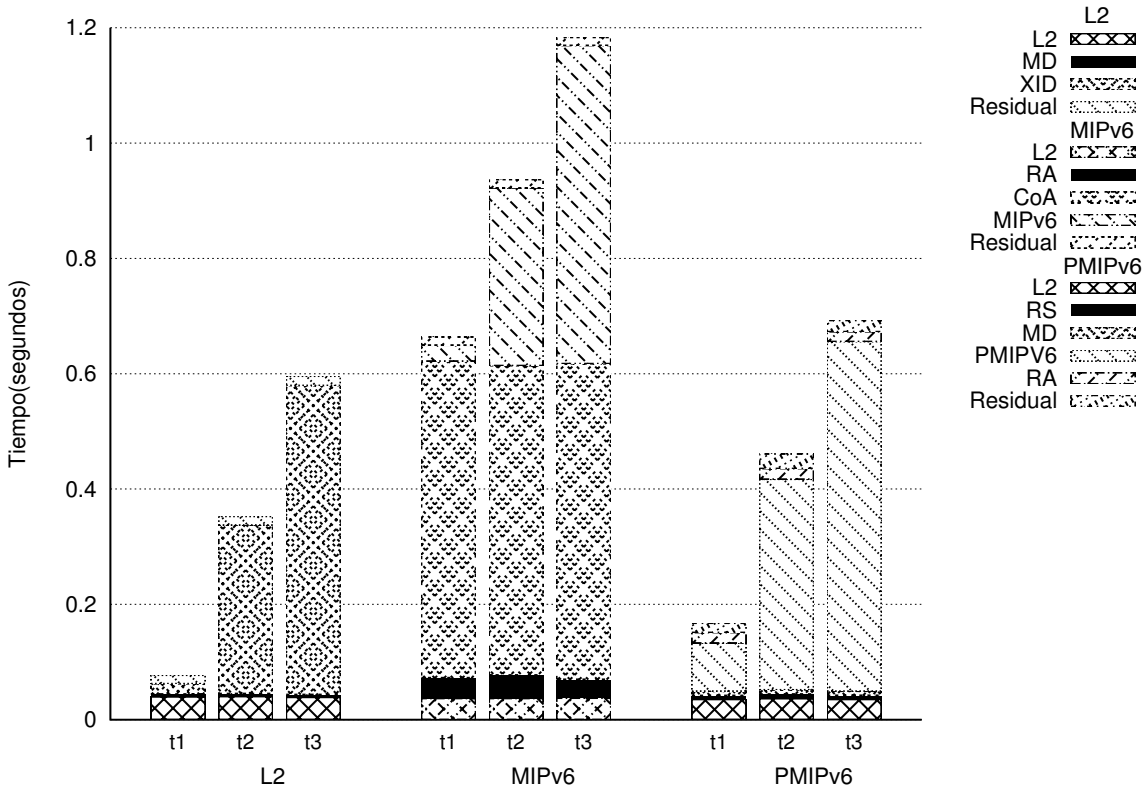

Figura 6.17: Histograma: Comparativa de las soluciones

En este histograma se puede ver como aumenta el tiempo total según se aumenta el retardo en el enlace entre los nodos inalámbricos y el servidor, y a que tiempo afecta en cada caso.

Se puede ver que comparando el tiempo total de traspaso, la peor solución siempre es MIPv6, mientras que la mejor es nivel del enlace.

Además, se puede observar que en todos ellos hay un tiempo residual, que principalmente es el tiempo entre los ping request que se envían desde el nodo corresponsal.

6.6. Conclusiones

Después de estudiar y analizar todas las soluciones se puede decir que la mejor opción en cuanto a tiempo de traspaso total, es la movilidad a nivel de enlace. Además no depende de ningún mensaje enviado por el nodo móvil, como en el caso de PMIPv6, que tiene que enviar un RS; o recibido, como en MIPv6, que para detectar el movimiento depende del tiempo entre RAs configurado en el AR. Sin embargo exige que todos los puntos de acceso pertenezcan a la misma subred, por lo que si el tamaño de la red es demasiado grande, no escalaría y se tendría que dividir en varias subredes, sin poder aplicar la movilidad a nivel de enlace que se ha explicado.

Aunque el tiempo total del protocolo MIPv6 es muy variable y lento, sigue siendo una solución atractiva ya que es el único que ofrece movilidad basada en en cliente, y que por

lo tanto, permite a un usuario seguir conectado independientemente de la red a la que se conecte. Además si el tiempo de configuración de la dirección CoA se mejorase, estaría en tiempos simulares a los de PMIPv6.

PMIPv6 obtiene resultados intermedios, no siendo tan rápida como la solución a nivel de enlace, pero sí más rápida que MIPv6, además no teniendo tanta variabilidad como ésta.

Parte IV

Conclusiones y trabajos futuros

Capítulo 7

Conclusiones y trabajos futuros

7.1. Conclusiones

El objetivo de este Trabajo Fin de Grado ha sido el estudio de diferentes soluciones de movilidad estandarizadas en este momento. Para este estudio se ha medido el tiempo total de traspaso o handover desde que el nodo móvil comienza su traspaso hasta que vuelve a recibir tráfico. Además el estudio se ha realizado sobre una red real desplegada.

El primer paso fue desplegar la red inalámbrica que se iba a usar. Esta red se ha desplegado por todo el Departamento de Ingeniería Telemática de la Universidad. Esta red está compuesta por 8 nodos inalámbricos y un servidor, que da acceso a Internet a los nodos y controla el estado de estos a través del programa Nagios.

En los nodo inalámbricos se instaló la versión Debian 6 Squeeze y se instalaron los paquetes y drivers necesarios para su funcionamiento. Realizando algunas pruebas en los nodos se vio que había problemas con la configuración del driver Madwifi de las tarjetas inalámbricas, ya que hacía que el comportamiento de los nodos fuera inestable y dejaran de responder. Debido a esto, se realizó la migración a ath5k, instalando todos los paquetes necesarios para su configuración, obteniendo los mismos resultados que con Madwifi pero sin que los nodos dejaran de responder.

Con los nodos configurados, se procedió a configurar dos redes inalámbricas de acceso con dos portales cautivos. Uno de los portales está abierto y sin ningún tipo de autenticación, mientras que el segundo, está restringido con un usuario y contraseña. El servidor realiza las tareas de autenticación de los usuarios, además de dar acceso a Internet a estos.

Para la configuración de estos portales fue necesario crear una red privada VPN, ya que el programa usado para dar este servicio, tenía que estar a un salto IP de los nodos. Esta solución es soportada tanto si se hace por el cable como si se hace utilizando la red mesh y los enlaces inalámbricos.

Para terminar el despliegue de la red, se instalaron físicamente los nodos en los despachos, con su cableado y sus antenas. Las antenas se usaron para conectar los nodos entre ellos, creando enlaces inalámbricos. Se analizó el comportamiento con diferentes antenas y configuraciones que dieran el mayor rendimiento posible en términos de ancho de banda. En la mayoría de los casos de usaron antenas de 5 GHz, ya que es una banda en

la que hay menor interferencias obteniendo el efecto deseado.

Con la red desplegada, se procedió a estudiar la movilidad en esta. En todos los casos se buscó que la movilidad fuera lo más rápida posible.

En primer lugar se estudió la movilidad a nivel de enlace en la red configurada con los portales cautivos. Se estudió la señalización que se producía viendo si se enviaba algún mensaje informando de donde se encontraba el nodo móvil. En este caso, el programa OpenVPN, utilizado para crear la red privada, lo enviaba haciendo que el nodo servidor reenviara el tráfico correctamente.

En segundo lugar, se estudió la movilidad a nivel de enlace con los protocolos MIPv6 y PMIPv6. En el caso de MIPv6 es un protocolo de movilidad basada en el cliente; mientras que PMIPv6 es un protocolo de movilidad basada en la red. En ambos casos fue necesario compilar el kernel 2.6.39.2 con las opciones necesarias para ejecutar las implementaciones de estos protocolos en las entidades que participaban en la movilidad.

En MIPv6, se instaló y configuró la implementación de UMIP. En esta implementación se dio un problema con el tiempo de configuración de la dirección CoA, ya que siempre variaba entre 50 ms y 1 s, sin encontrar la razón por la que ocurría esto.

En PMIPv6, se configuró la implementación desarrollada por EURECOM. Esta no cumplía estrictamente con el estándar, ya que no enviaba los mensajes como se estandariza en la RFC [LJW11], pero su funcionamiento era el correcto.

Finalmente se introdujo un retardo entre los nodos inalámbricos y el servidor para ver su efecto en la movilidad, simulando un tamaño de red mayor entre ellos. Para ello se utilizó la herramienta netem y un PC intermedio que actuaba de router y era el que introducía el retardo. Además se configuraron las direcciones IP y las rutas necesarias para que el tráfico se enviara al PC.

Teniendo la red configurada se procedió a tomar medidas de todas las soluciones para poder analizarlas y compararlas.

De los resultados obtenidos se puede concluir que la mejor opción en cuanto al menor tiempo de traspaso total medido es la solución a nivel de enlace, seguido de PMIPv6 y MIPv6. Además, como se ha comentado, el tiempo medido en MIPv6 es muy variable.

7.2. Trabajos futuros

- En cuanto a la red inalámbrica desplegada en el departamento de Ingeniería Telemática se podría mejorar en los siguientes aspectos:
  - Definir diferentes tipos de usuarios (premium, estándar) con diferentes características, como por ejemplo, limitar la velocidad, limitar el número de MB descargados, etc.
  - Conectar el servidor de autenticación FreeRADIUS a una base de datos, en lugar de que los usuarios estén en un fichero de texto plano. Esa base de datos se podrían crear o se podría conectar la base de datos de usuarios del departamento de la universidad.
- En MIPv6 mejorar el tiempo de configuración de la dirección IP CoA, intentando mejorar la implementación o programando una nueva.

- Mejorar el tiempo de detección en MIPv6 escuchando mensajes de nivel de enlace, en lugar de esperar a que el router de acceso envíe un RA.
- Estudiar la movilidad de MIPv6 en trapasos desde una red celular a una red WiFi y viceversa.
- En PMIPv6 se puede mejorar la implementación en estos aspectos:
  - Hacer que la implementación cumpla con el estándar definido en la RFC, que como se ha visto en la figura 6.9, que cumple en cuanto a funcionalidad pero no en cuanto a la señalización.
  - Se podría mejorar el tiempo de detección del nodo móvil cuando cambia de MAG, escuchando mensajes de nivel de enlace sin esperar que el nodo móvil envíe un RS.
  - Mejorar el tiempo de configuración del túnel. En los *logs* de la implementación se ha visto que elimina el túnel y lo vuelve a crear, en lugar de modificarlo. Además, como se ha visto el tiempo ronda unos 80ms cuando no hay ningún tiempo de retardo, mientras que en MIPv6 ese tiempo es de 30ms. La funcionalidad en ambos casos es igual, ya que es modificar el túnel y la ruta hacia el nodo móvil, por lo que es un tiempo que se puede rebajar.
- Estudiar la movilidad a nivel de enlace, MIPv6 y PMIPv6 en redes celulares.
- Poner en producción alguno de los mecanismos de movilidad con usuarios reales, analizando el comportamiento con trazas de tráfico y con encuestas a los usuarios.

Parte V

Anexos

Apéndice A

Planificación de tareas y presupuesto

A.1. Introducción

A.2. Descomposición en tareas

En esta sección se va a presentar la descomposición de tareas que se han llevado a cabo para realizar este Trabajo Fin de Grado.

Se describen cada una de las tareas así como la relación con otras tareas y el esfuerzo de cada una.

- **Tarea A: Despliegue de la red inalámbrica**
  - **Subtarea A.1:** Estudio de los nodos Saxnet Meshnode III.
    - **Descripción:** en esta tarea se estudia el equipamiento que se va a usar, para conocer las características del mismo.
    - **Objetivos:** conocer las características del equipamiento.
    - **Relación con otras tareas:** esta tarea da comienzo al proyecto.
    - **Duración:** 2 semanas
    - **Recursos:** Ingeniero 0.25 ingenieros/mes.
  - **Subtarea A.2:** Instalación de Debian en los nodos inalámbricos
    - **Descripción:** en esta tarea se instala la distribución Debian 6 Squeeze en los nodos inalámbricos Saxnet Meshnode III. Para ello se crea una nueva partición y se instala en ese partición.
    - **Objetivos:** se pretende que todos los nodo inalámbricos tengan una instalación personalizada de Debian, que no dependa del fabricante de los nodos.
    - **Relación con otras tareas:** comienza después de la tarea A.1.
    - **Duración:** 2 semanas
    - **Recursos:** Ingeniero 0.5 ingenieros/mes.
  - **Subtarea A.3:** Compilar e instalar kernel en los nodos inalámbricos
    - **Descripción:** en esta tarea se compila un kernel para los nodos inalámbricos.

- **Objetivos:** se pretende que todos los nodos tengan este kernel instalado.
- **Relación con otras tareas:** esta tarea comienza después de la tarea A.2.
- **Duración:** 2 semanas
- **Recursos:** Ingeniero 0.5 ingenieros/mes.

- **Subtarea A.4:** Configurar el software en los nodos inalámbricos y en el servidor
  - **Descripción:** en esta tarea se instalan todos los paquetes necesarios para el funcionamiento de los drivers, se configuran los nodos para que su memoria principal sea en modo lectura, etc. Además se instala y configura el programa Nagios en el servidor.
  - **Objetivos:** se pretende tener todos los nodos que se van a desplegar configurados correctamente y que el servidor tenga configurado Nagios.
  - **Relación con otras tareas:** esta tarea comienza después de la tarea A.3.
  - **Duración:** 5 semanas
  - **Recursos:** Ingeniero 1.25 ingenieros/mes.
- **Subtarea A.5:** Diseño de la red
  - **Descripción:** se asignan las diferentes direcciones IP que va a tener cada nodo y el servidor, así como el routing necesario.
  - **Objetivos:** diseñar la red con las direcciones IPs necesarias.
  - **Relación con otras tareas:** esta tarea dará comienzo después de la tarea A.4.
  - **Duración:** 1 semana.
  - **Recursos:** Ingeniero 0.25 ingenieros/mes.
- **Subtarea A.6:** Portal cautivo
  - **Descripción:** en esta tarea se estudian los programas para crear portales cautivos, se elige el que mejor cumple con los requisitos. Finalmente se instala y se configura tanto en el servidor como en los nodos inalámbricos para tener dos portales cautivos.
  - **Objetivos:** se quiere tener configurado y operativo dos portales cautivo diferentes en la red inalámbrica desplegada.
  - **Relación con otras tareas:** esta tarea comenzará tras la tarea A.5.
  - **Duración:** 5 semanas.
  - **Recursos:** Ingeniero 1.25 ingenieros/mes.
- **Subtarea A.7:** Despliegue físico de los nodos
  - **Descripción:** se colocan los nodos en los despachos, conectando todos los cables necesarios y se conectan las antenas apuntando a los nodos cercanos.
  - **Objetivos:** se espera tener la red desplegada para realizar las pruebas.
  - **Relación con otras tareas:** esta tarea comienza tras la tarea A.6.
  - **Duración:** 4 semanas.
  - **Recursos:** Ingeniero 1 ingeniero/mes.

- **Tarea B: Documentación y análisis del estado del arte**

- **Subtarea B.1:** Estudio de los diferentes niveles de movilidad.
  - **Descripción:** se estudian los niveles de la pila TCP/IP en lo que existen soluciones de movilidad desarrolladas.
  - **Objetivos:** se pretende conocer las diferentes opciones que hay.

- **Relación con otras tareas:** esta tarea da comienzo tras la tarea A.
- **Duración:** 1 semana
- **Recursos:** Ingeniero 0.125 ingenieros/mes.

- **Subtarea B.2:** Estudio de la movilidad a nivel de enlace y nivel de red.
  - **Descripción:** se estudian las soluciones de nivel de enlace en WiFi y redes celulares, y los protocolos MIPv6 y PMIPv6 de nivel de red.
  - **Objetivos:** conocer el funcionamiento y la señalización de las soluciones y protocolos.
  - **Relación con otras tareas:** esta tarea comienza después de la tarea B.1.
  - **Duración:** 1 semana
  - **Recursos:** Ingeniero 0.125 ingenieros/mes.

- **Tarea C: Nivel de enlace**
  - **Subtarea C.1:** Estudio de señalización
    - **Descripción:** se estudia los mensajes que se envían y los distintos tiempos que se pueden medir para determinar el tiempo total del handover.
    - **Objetivos:** determinar los mensajes que se intercambian para desarrollar los scripts necesarios.
    - **Relación con otras tareas:** esta tarea da comienzo después de la tarea B.
    - **Duración:** 1 semana.
    - **Recursos:** Ingeniero 0.125 ingenieros/mes.
  - **Subtarea C.2:** Desarrollo de un script de automatización de medidas
    - **Descripción:** durante este tarea se desarrolla un script en bash para medir el traspaso total.
    - **Objetivos:** tener un script que mide el traspaso total en la solución de nivel de enlace.
    - **Relación con otras tareas:** esta tarea comienza después de la tarea C.1.
    - **Duración:** 3 semanas.
    - **Recursos:** Ingeniero 0.375 ingenieros/mes.

- **Tarea D: MIPv6**
  - **Subtarea D.1:** Instalación y configuración
    - **Descripción:** en esta tarea se instala el kernel y el programa MIPv6 en las entidades que participan en la señalización. Además se crea al archivo de configuración con las opciones necesarias y se configuran los router de acceso y el agente local para que envían Router Advertisement.
    - **Objetivos:** se espera tener configurado MIPv6 y funcionando correctamente.
    - **Relación con otras tareas:** esta tarea da comienzo después de la tarea C.
    - **Duración:** 2 semanas.
    - **Recursos:** Ingeniero 0.25 ingenieros/mes.
  - **Subtarea D.2:** Estudio de la señalización

- **Descripción:** se estudia la señalización intercambiada en MIPv6, verificando que se produce correctamente. Además durante esta tarea se intenta que el tiempo de configuración de la dirección CoA sea menor.
- **Objetivos:** conocer los mensajes intercambiados para poder desarrollar el script de automatización.
- **Relación con otras tareas:** esta tarea comienza tras la tarea D.1.
- **Duración:** 5 semanas.
- **Recursos:** Ingeniero 0.75 ingenieros/mes.

- **Subtarea D.3:** Desarrollo de un script de automatización de medidas
  - **Descripción:** desarrollar un script en bash que mida los tiempos intermedios y el tiempo del traspaso total.
  - **Objetivos:** tener un script que mida los tiempos.
  - **Relación con otras tareas:** esta tarea comienza después de la tarea D.2.
  - **Duración:** 2 semanas.
  - **Recursos:** Ingeniero 0.125 ingenieros/mes.

- **Tarea E: PMIPv6**

- **Subtarea E.1:** Instalación y configuración
  - **Descripción:** se instalan el kernel y el programa en las entidades que participan en la señalización. También se configura el programa con las opciones necesarias.
  - **Objetivos:** tener PMIPv6 configurado y funcionando.
  - **Relación con otras tareas:** esta tarea comienza tras la tarea D.
  - **Duración:** 2 semanas.
  - **Recursos:** Ingeniero 0.25 ingenieros/mes.
- **Subtarea E.2:** Estudio de la señalización
  - **Descripción:** se estudia la señalización intercambiada en PMIPv6, verificando que se produce correctamente.
  - **Objetivos:** conocer los mensajes intercambiados para poder desarrollar el script de automatización.
  - **Relación con otras tareas:** esta tarea comienza después de la tarea E.1.
  - **Duración:** 2 semanas.
  - **Recursos:** Ingeniero 0.25 ingenieros/mes.
- **Subtarea E.3:** Desarrollo de un script de automatización de medidas
  - **Descripción:** desarrollar un script en bash que mida los tiempos intermedios y el tiempo del traspaso total.
  - **Objetivos:** tener un script que mida los tiempos.
  - **Relación con otras tareas:** esta tarea da comienzo tras la tarea E.2.
  - **Duración:** 2 semanas.
  - **Recursos:** Ingeniero 0.125 ingenieros/mes.

- **Tarea F: Medidas**

- **Subtarea F.1:** Configuración de simulación del retardo.
  - **Descripción:** se estudia la configuración que necesita el comando netem y se crea un script de automatización. Además se configura el routing necesario para que los paquetes atraviesen el router intermedio.

- **Objetivos:** se espera tener la red y todos los scripts necesarios configurados.
- **Relación con otras tareas:** esta tarea da comienzo tras la tarea E.
- **Duración:** 1 semana.
- **Recursos:** Ingeniero 0.125 ingenieros/mes.

- **Subtarea F.2:** Toma de medidas
  - **Descripción:** en esta tarea se ejecutan los scripts desarrollados en todos los protocolos.
  - **Objetivos:** se espera tener medidos todos los tiempos del traspaso en todos los protocolos.
  - **Relación con otras tareas:** esta tarea da comienzo tras la tarea F.1.
  - **Duración:** 1 semana.
  - **Recursos:** Ingeniero 0.125 ingenieros/mes.

- **Tarea G: Resultados**
  - **Tarea G.1:** Evaluación de los resultados
    - **Descripción:** en esta tarea se estudia los resultados obtenidos y se dibujan las gráficas con el programa gnuplot, que luego se usarán en la memoria.
    - **Objetivos:** analizar los datos y dibujar las figuras para la memoria.
    - **Relación con otras tareas:** esta tarea comienza tras la tarea F.
    - **Duración:** 2 semanas.
    - **Recursos:** Ingeniero 0.25 ingenieros/mes.

- **Tarea H: Memoria**
  - **Subtarea H.1:** Organización y estructura de la memoria
    - **Descripción:** se organiza la memoria y la estructura que va a tener.
    - **Objetivos:** organizar y estructurar la memoria.
    - **Relación con otras tareas:** esta tarea comienza tras la tarea G.
    - **Duración:** 1 semana.
    - **Recursos:** Ingeniero 0.125 hombres/mes.
  - **Subtarea H.2:** Redacción de la memoria
    - **Descripción:** durante esta tarea se redacta el documento.
    - **Objetivos:** redactar el documento con sus capítulos y anexos.
    - **Relación con otras tareas:** esta tarea da comienzo tras la tarea H.1.
    - **Duración:** 6 semanas.
    - **Recursos:** Ingeniero 0.75 ingenieros/mes.
  - **Subtarea H.3:** Redacción del resument
    - **Descripción:** durante esta tarea se redacta el resumen de la memoria.
    - **Objetivos:** redactar el documento resumen de la memoria
    - **Relación con otras tareas:** esta tarea da comienzo tras la tarea H.2.
    - **Duración:** 1 semana.
    - **Recursos:** Ingeniero 0.125 ingenieros/mes.

Tarea	Duración (semanas)	Recursos (Ing/m)
Despliegue de la red inalámbrica		
A.1 Estudio de los nodos Saxnet Meshnode III	2	0.25
A.2 Instalación de Debian en los nodos inalámbricos	2	0.5
A.3 Compilar e instalar kernel en los nodos inalámbricos	2	0.5
A.4 Configuración del software en los nodos inalámbricos y servidor	5	1.25
A.5 Diseño de la red	1	0.25
A.6 Portal cautivo	5	1.25
A.7 Despliegue físico de los nodos	4	1
Total		**4**
Documentación y análisis del estado del arte		
B.1 Estudio de los diferentes niveles de movilidad	1	0.125
B.2 Estudio de la movilidad a nivel de enlace y nivel de red	1	0.125
Total		**0.25**
Nivel de enlace		
C.1 Estudio de señalización	1	0.125
C.2 Desarrollo de un script de automatización de medidas	3	0.375
Total		**0.5**
MIPv6		
D.1 Instalación y configuración	2	0.25
D.2 Estudio de la señalización	5	0.75
D.3 Desarrollo de un script de automatización de medidas	2	0.125
Total		**1.125**
PMIPv6		
E.1 Instalación y configuración	2	0.25
E.2 Estudio de la señalización	2	0.25
E.3 Desarrollo de un script de automatización de medidas	2	0.125
Total		**0.625**
Medidas		
F.1 Configuración de simulación del retardo	1	0.125
F.2 Toma de medidas	1	0.125
Total		**0.25**
Resultados		
G.1 Evaluación de los resultados	2	0.25
Total		**0.25**
Memoria		
H.1 Organización y estructura de la memoria	1	0.125
H.2 Redacción de la memoria	6	0.75
H.3 Redacción del resumen	1	0.125
Total		**1**
Total		**8**

Tabla A.1: Resumen descomposición en tareas

A.3. Planificación con el diagrama de fases de ejecución detallado

En la figura A.1 se muestra el diagrama de Gantt con las tareas principales. Además en la figura A.2 se presenta el diagrama de Gantt con todas las tareas que se han realizado en este Trabajo Fin de Grado.

Id	Nombre de tarea	Duración	Comienzo	Fin
1	**Tarea A: Despliegue de la red inalámbrica**	**104 días**	**lun 01/08/11**	**jue 22/12/11**
9	**Tarea B: Documentación y análisis del estado del arte**	**11 días**	**lun 23/01/12**	**lun 06/02/12**
12	**Tarea C: Nivel de enlace**	**21 días**	**lun 06/02/12**	**lun 05/03/12**
15	**Tarea D: MIPv6**	**41 días**	**lun 05/03/12**	**lun 30/04/12**
19	**Tarea E: PMIPv6**	**31 días**	**lun 30/04/12**	**lun 11/06/12**
23	**Tarea F: Medidas**	**11 días**	**lun 11/06/12**	**lun 25/06/12**
26	**Tarea G: Resultados**	**11 días**	**lun 25/06/12**	**lun 09/07/12**
28	**Tarea H: Memoria**	**41 días**	**lun 09/07/12**	**lun 03/09/12**

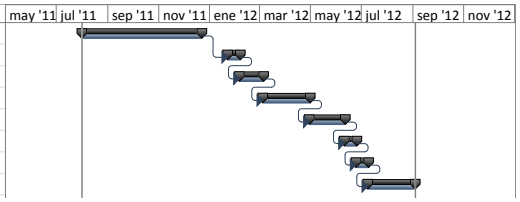

Figura A.1: Diagrama de Gantt con la planificación del proyecto resumida

Id	Nombre de tarea	Duración	Comienzo	Fin
1	**Tarea A: Despliegue de la red inalámbrica**	**104 días**	**lun 01/08/11**	**jue 22/12/11**
2	Tarea A.1: Estudio de los nodos Saxnet Meshnode III	11 días	lun 01/08/11	lun 15/08/11
3	Tarea A.2: Instalación de Debian en los nodos inalámbricos	11 días	lun 29/08/11	lun 12/09/11
4	Tarea A.3: Compilar e instalar kernel en los nodos inalámbricos	6 días	lun 12/09/11	lun 19/09/11
5	Tarea A.4: Configuración del software en los nodos inalámbricos y servidor	21 días	lun 19/09/11	lun 17/10/11
6	Tarea A.5: Diseño de la red	6 días	lun 17/10/11	lun 24/10/11
7	Tarea A.6: Portal cautivo	26 días	lun 24/10/11	lun 28/11/11
8	Tarea A.7: Despliegue físico de los nodos	19 días	lun 28/11/11	jue 22/12/11
9	**Tarea B: Documentación y análisis del estado del arte**	**11 días**	**lun 23/01/12**	**lun 06/02/12**
10	Tarea B.1: Estudio de la movilidad en los distintos niveles	6 días	lun 23/01/12	lun 30/01/12
11	Tarea B.2: Estudio de la movilidad a nivel de enlace y nivel de red	6 días	lun 30/01/12	lun 06/02/12
12	**Tarea C: Nivel de enlace**	**21 días**	**lun 06/02/12**	**lun 05/03/12**
13	Tarea C.1: Estudio de señalización	6 días	lun 06/02/12	lun 13/02/12
14	Tarea C.2: Desarrollo de un script de automatización de medidas	16 días	lun 13/02/12	lun 05/03/12
15	**Tarea D: MIPv6**	**41 días**	**lun 05/03/12**	**lun 30/04/12**
16	Tarea D.1: Instalación y configuración	11 días	lun 05/03/12	lun 19/03/12
17	Tarea D.2: Estudio de la señalización	21 días	lun 19/03/12	lun 16/04/12
18	Tarea D.3: Desarrollo de un script de automatización de medidas	11 días	lun 16/04/12	lun 30/04/12
19	**Tarea E: PMIPv6**	**31 días**	**lun 30/04/12**	**lun 11/06/12**
20	Tarea E.1: Instalación y configuración	11 días	lun 30/04/12	lun 14/05/12
21	Tarea E.2: Estudio de la señalización	11 días	lun 14/05/12	lun 28/05/12
22	Tarea E.3: Desarrollo de un script de automatización de medidas	11 días	lun 28/05/12	lun 11/06/12
23	**Tarea F: Medidas**	**11 días**	**lun 11/06/12**	**lun 25/06/12**
24	Tarea F.1: Configuración de simulación del retardo	6 días	lun 11/06/12	lun 18/06/12
25	Tarea F.2: Toma de medidas	6 días	lun 18/06/12	lun 25/06/12
26	**Tarea G: Resultados**	**11 días**	**lun 25/06/12**	**lun 09/07/12**
27	Tarea G.1: Evaluación de los resultados	11 días	lun 25/06/12	lun 09/07/12
28	**Tarea H: Memoria**	**41 días**	**lun 09/07/12**	**lun 03/09/12**
29	Tarea H.1: Organización y estructura de la memoria	6 días	lun 09/07/12	lun 16/07/12
30	Tarea H.2: Redacción de la memoria	31 días	lun 16/07/12	lun 27/08/12
31	Tarea H.3: Redacción del resumen	6 días	lun 27/08/12	lun 03/09/12

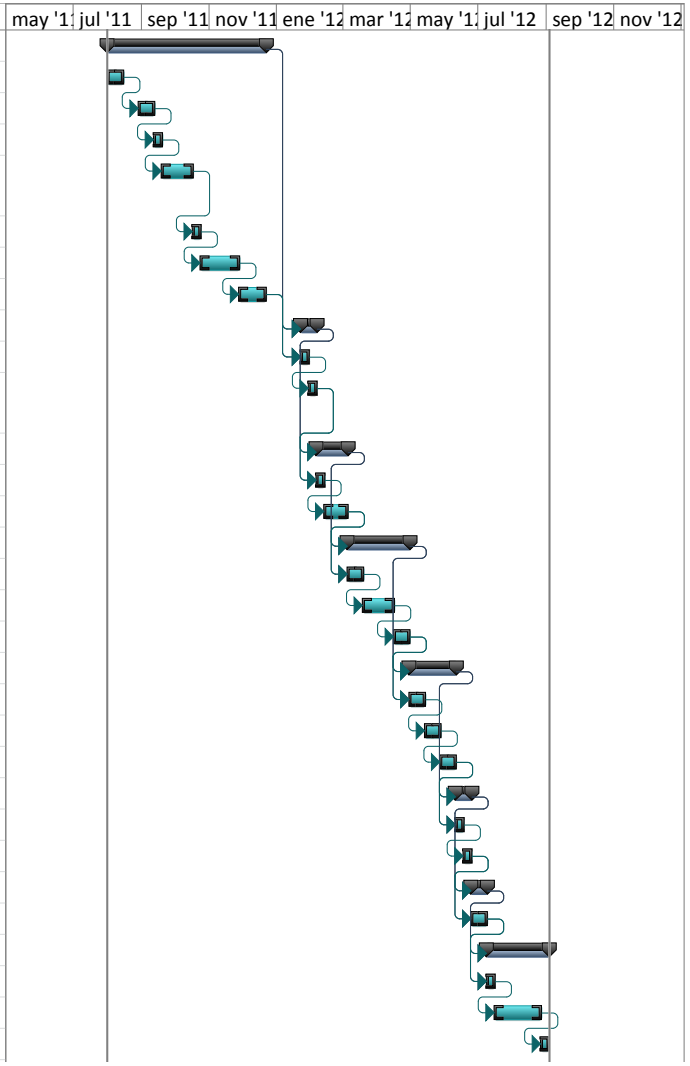

Figura A.2: Diagrama de Gantt con la planificación detallada del proyecto

A.4. Recursos

En esta sección se distinguen los diferentes recursos usados para la realización del Trabajo Fin de Grado:

- Recursos materiales:
  - Ordenadores de sobremesa.
    - 2 PCs *Intel*® *Core*^{TM} 2 Quad Processor Q9400 @ 2.66GHz, 4GB RAM, Sistem Operativo Debian 6 Squeeze
    - 1 PC *Intel*® *Pentium*® D CPU 3.20GHz, 2GB RAM, Sistema Operativo: Debian 6 Squeeze
  - 9 Nodos inalámbricos Saxnet Meshnode III (incluidos cables de red y antenas).
- Recursos de trabajo:
  - 1 Graduado en Ingeniería Telemática: 6 ingenieros/mes
  - 1 Graduado en Ingeniería Telemática: 2 ingenieros/mes
  - 2 Ingenieros Senior: 0.5 ingenieros/mes
- Otros costes:
  - Conexión a internet durante 13 meses.

A.5. Presupuesto de Proyecto

1. Autor: Miriam Marciel Noguera
2. Departamento: Ingeniería Telemática
3. Descripción del Proyecto:
   - Título: Estudio de soluciones de movilidad a nivel de enlace y nivel de red.
   - Duración: 13 meses
   - Tasa de costes indirectos: 20 %.
4. Presupuesto total del Proyecto (valorado en Euros): euros. Ver tabla A.2
5. Subcontratación de tareas: no se especifican.
6. Otros costes directos del proyecto: no se especifican.

Concepto	**Cantidad (€)**	**Coste €**	**% Proyecto**	**Dedicación (meses)**	**Depreciación (meses)**	**Total €**
Recursos materiales						
Ordenadores de sobremesa	3	550	100	13	60	357,5
Nodo inalámbricos	9	3513	100	13	120	3.425,18
Total						**3.782,68**
Recursos de trabajo						
Graduado en Ing. Telemática	1 (6 ing/mes)	2.694,39	-	-	-	16.166,34
Graduado en Ing. Telemática	1 (2 ing/mes)	2.694,39	-	-	-	5.388,78
Ingenieros Senior	2 (0.5 ing/mes)	4.289,54	-	-	-	2.144,77
Total						**23.699,89**
Otros costes						
Conexión a internet	1	30	-	13	-	390
Total						**390**
Total						**27.872,57 €**

Tabla A.2: Tabla presupuesto

Apéndice B

Configuración del servidor

B.1. Introducción

En este apéndice se describen las características del servidor y la configuración necesaria para crear el portal cautivo, el acceso a Internet de los nodos y de lo usuarios y la configuración de Nagios.

B.2. Características del equipo

- CPU: $Intel^{®}$ $Core^{TM}$ 2 Quad Processor Q9400 @ 2.66GHz
- Disco duro: 320 GB
- RAM: 4 GB
- RTL8111/8168B PCI Express Gigabit Ethernet

B.3. Configuración

B.3.1. Configuración del portal cautivo

Para la configuración del portal cautivo se necesitaba que los puntos de acceso estuvieran a un salto IP, lo que se consiguió creando una red virtual privada (OpenVPN). Además el portal cautivo necesitaba un servidor web (Apache) y un servidor de autenticación (FreeRADIUS), como se ha explicado anteriormente.

B.3.1.1. OpenVPN

Para instalar y configurar el programa se siguieron los siguientes pasos:

1. Instalar los siguientes paquetes:

```
apt-get install openvpn bridge-utils
```

2. Generar los certificados de seguridad para la configuración de los túneles.
   El primer certificado necesario es el certificado de autoridad (CA), que se genera con los siguientes comandos:

   ```
   cd /etc/openvpn/
   . ./vars
   ./clean-all
   ./build-ca
   ```

   Después se crearon los certificados del servidor:

   ```
   ./build-key-server server
   ```

   Y por último los certificados para cada uno de los nodos:

   ```
   ./build-key meshnode01
   ```

   Donde meshnode01 se cambia por cada uno de los nodos. Cada uno de estos certificados hay que copiarlos a los clientes además del certificado de autoridad.

3. Ejecutar los scripts `bridge-start_auth.sh` y `bridge-start_no_auth.sh`. Estos scripts crean la interfaz `tap` y se puentea con la interfaz *dummy*.

4. Configurar los archivos `server_auth.conf` y `server_no_auth.conf` con las opciones necesarias. Entre ellas destaca la dirección IP del túnel, el puerto y protocolo; y el tipo de VPN.

5. Por último se lanza la aplicación con el comando:

   ```
   /etc/init.d/openvpn start
   ```

Con estos pasos se tiene configurada una VPN en el servidor a la que los nodos se tendrán que conectar. Mencionar que toda esta configuración tiene que duplicarse para crear las dos redes.

B.3.1.2. Apache

El programa Apache se instaló para que el portal cautivo mostrara la página de autenticación del usuario. Se siguieron los siguientes pasos para su configuración:

1. Instalación del programa

   ```
   apt-get install apache2 libapache2-mod-php5 libssl-dev
   ```

2. Se generaron los certificados SSL para garantizar la identidad del servidor:

   ```
   make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/apache2/key.pem
   ```

3. Se crea el fichero de configuración del servidor en la carpeta `/etc/apache2/sites-available/chillispot`.

4. Añadir los puertos para permitir que el servidor escuche en el puerto HTTPS. Para ello se añade en el fichero `/etc/apache2/ports.conf` la siguiente configuración:

```
<IfModule mod_ssl.c>
   Listen 443
</IfModule>
```

5. Activar la configuración que se acaba de crear y arrancar el servidor:

```
a2ensite chillispot
/etc/init.d/apache2 start
```

B.3.1.3. FreeRADIUS

La aplicación FreeRADIUS fue necesaria para autenticar a los usuarios cuando se conectan a la red. Para su instalación se siguieron los siguientes pasos:

1. Para instalar FreeRadius se descargaron las fuentes y se compilaron con los siguientes pasos:

```
wget ftp://ftp.freeradius.org/pub/freeradius/freeradius-server-2.1.12.tar.gz
tar -xzvf freeradius-server-2.1.12.tar.gz
cd freeradius-server-2.1.12
make
make install
```

2. Se configura el servidor Radius en la misma máquina y en su dirección de localhost. Para ello se configura el archivo `radiusd.conf`.

3. Configurar el archivo `clients.conf` para que la única IP de la que recibiera peticiones fuera de la IP de localhost:

```
##
## clients.conf - client configuration directives
##

client 127.0.0.1{
       ipaddr = 127.0.0.1
       secret = CONTRASEÑA
       shortname = localhost
       nastype = other
}
```

4. Añadir los usuarios al archivo `users` con su nombre de usuario y su contraseña de la siguiente manera:

```
nombreusuario Cleartext-Password := "contraseña"
```

5. Con esta configuración ya se puede lanzar el servidor Radius:

```
radiusd
```

B.3.1.4. Chillispot

El último paso para tener el portal cautivo operando es instalar y configurar Chillispot. Para su instalación y configuración se siguieron estos pasos:

1. Instalar el programa con los comandos:

   ```
   wget http://www.chillispot.info/download/chillispot_1.0_i386.deb
   dpkg -i -force-architecture chillispot_1.0_i386.deb
   ```

   En un primer momento se instaló el programa de las fuentes, pero se vio que no era compatible con los PCs de 64 bits. Por ello se hace la instalación del .deb forzando la arquitectura de 32 bits.

2. Crear los ficheros de configuración `chilli_auth.conf` y `chilli_no_auth.conf`. Principalmente hubo que configurar el rango del servidor DHCP, el servidor de DNS, los parámetros para la configuración del servidor Radius y del servidor de autenticación.

3. Crear unas reglas con iptables para que creara una red NAT en la que los usuarios y los nodos inalámbricos utilizaran la misma dirección IP pública en Internet en el fichero chilli.iptables:

4. Crear el fichero `/usr/lib/cgi-bin/hotspotlogin.cgi` que incluye el código HTML con la página que se muestra al usuario, además de todas las operaciones para comunicarse con el servidor de autenticación. En el caso de la red que está protegida con un usuario y contraseña, se dejó el fichero por defecto. Sin embargo la red sin ningún tipo de restricción hubo que cambiar el fichero para que mostrara otra página y no pidiera un nombre de usuario y una contraseña, sino sólo unos términos y condiciones y un botón de Aceptar. Las modificaciones de este último caso se muestran a continuación marcadas en rojo:

   ```
   138 #If attempt to login
   139 if ($button =~ /Login$/) {
   140         $username = user;
   141         $password = pass;
   142         $hexchal = pack "H32", $challenge;
   ```

5. Por último se lanza el programa indicando el fichero que tiene que leer:

   ```
   chilli -conf /etc/chilli_auth.conf
   ```

B.3.2. Nagios

B.3.2.1. Instalación

Para la instalación del software de monitorización de los nodos inalámbricos Nagios, primero se instalaron los paquetes necesarios para su funcionamiento:

```
apt-get install apache2 libapache2-mod-php5 libjpeg62 libjpeg62-dev
```

```
libpng12-0 libpng12-dev build-essential
```

Después se creó una cuenta de usuario para Nagios necesario para la ejecución:

```
/usr/sbin/useradd -m -s /bin/bash nagios
passwd nagios
/usr/sbin/groupadd nagcmd
/usr/sbin/usermod -a -G nagcmd nagios
/usr/sbin/usermod -a -G nagcmd www-data
```

Con los paquetes necesarios y el usuario creado, se descargó e instaló el programa:

```
wget http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-3.2.3.tar.gz
tar xzf nagios-3.2.3.tar.gz
cd nagios-3.2.3
./configure -with-command-group=nagcmd
make all
make install
make install-init
make install-config
make install-commandmode
```

Después se procedió a la configuración de la interfaz web, instalando el complemento y creando un usuario y contraseña para acceder a ella:

```
make install-webconf
htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

Por último se instalaron los plugins de Nagios y se lanza el programa:

```
wget http://prdownloads.sourceforge.net/sourceforge/nagiosplug/
nagios-plugins-1.4.11.tar.gz
tar xzf nagios-plugins-1.4.11.tar.gz
cd nagios-plugins-1.4.11
./configure -with-nagios-user=nagios -with-nagios-group=nagios
-with-ping-command=ping
make
make install
/etc/init.d/nagios start
```

B.3.2.2. Configuración

En la configuración de Nagios existen varios ficheros, que se encuentran en `/usr/local/nagios/etc/`. Al cambiar la configuración de los archivos se aconseja ejecutar el siguiento comando para ver si hay algún fallo en alguno de los estos:

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

Primero se va a comentar las modificaciones para configurar las alertar por email en caso de que alguno de los nodos fallara:

1. Dentro del fichero `objects/commands.cfg`, se creó el comando a ejecutar:

```
define command{
       command_name netc-mail-host
       command_line /tmp/correo.pl $CONTACTEMAIL$
       $NOTIFICATIONTYPE$ $HOSTNAME$ $HOSTSTATE$ $HOSTADDRESS$
}
```

   Donde el fichero `/tmp/correo.pl` tiene las siguientes instrucciones:

```
#!/usr/bin/perl
use Net::SMTP;
$destinatario=$ARGV[0];
$tipo=$ARGV[1];
$host=$ARGV[2];
$estado=$ARGV[3];
$ip=$ARGV[4];
$smtp= Net::SMTP-> new ("smtp.uc3m.es",Debug=>1);
$smtp->mail("nagios it.uc3m.es");
$smtp->to("$destinatario");
$smtp->data();
$smtp->datasend("To: $destinatario\n");
$smtp->datasend("Subject: NAGIOS - $tipo: $host con estado
$estado\n");
$smtp->datasend("Notificacion de tipo: $tipo\n");
$smtp->datasend("Equipo: $host ($ip)\n");
$smtp->datasend("Estado actual: $estado\n");
$smtp->datasend();
$smtp->quit;
```

2. Se crean los contactos a los que se van a enviar las alertas en el fichero `objects/contacts.cfg`. Primero se crea un contacto genérico en el que se van a definir las horas de contactos, los servicios y los hosts:

```
define contact{
       name netc
       service_notification_period 24x7 ;workhours
       host_notification_period 24x7 ;workhours
       service_notification_options w,u,c,r,f,s
       host_notification_options d,u,r,f,s
       service_notification_commands netc-mail-host ;notify-service-by-email
       host_notification_commands netc-mail-host
       register 0
}
```

   Después se definen todos los contactos para que hereden la configuración del genérico que se ha creado. Se define un nombre y la dirección de correo:

```
define contact {
        contact_name miriam
        use netc
        alias miriam
        email 100072751@alumnos.uc3m.es
}
```

Por último se añade un grupo de contactos, en el que se añaden todos los miembros. Esto no es necesario si sólo se tiene un usuario:

```
define contactgroup{
        contactgroup_name admins
        alias Administradores
        members miriam, usuario1
}
```

Con esta configuración ya se tiene definido los contactos, con lo que falta añadir los nodos a los que se tiene que monitorizar de la siguiente manera:

1. Se crean grupos de hosts. En nuestra configuración se crearon dos grupos según la red en la que estaban conectados. Para crearlos hay que modificar el fichero `objects/switch.cfg` de la siguiente manera:

   ```
   define hostgroup{
           hostgroup_name meshnode139
           alias meshnode139
   }
   ```

2. Se añaden los nodos inalámbricos que tiene que monitorizar. Para ello, hay que modificar el fichero `objects/switch.cfg` añadiendo el siguiente texto:

   ```
   define host{
           use generic-switch
           host_name meshnode01
           alias meshnode01
           address 10.0.139.101
           hostgroups meshnode139
           notification_interval 0
   }
   ```

3. Por último se añade el servicio que se quiere monitorizar en cada nodo. En nuestro caso el servicio es un ping. Además se especifica cada cuanto tiempo se envía y el grupo al que alertar en caso de que cambie el estado.

   ```
   define service{
           use generic-service
           host_name meshnode01
           service_description PING
           check _command check_ping!200.0,20%!600.0,60%
           normal_check_interval 0.1
           retry_check_interval 0.1
           contact_groups admins
   }
   ```

Apéndice C

Instalación y configuración de los nodos inalámbricos

C.1. Introducción

En este apéndice se describen las características detalladas de los equipos Saxnet Meshnode III, la instalación y la configuración para crear la red Wi-Fi.

C.2. Características

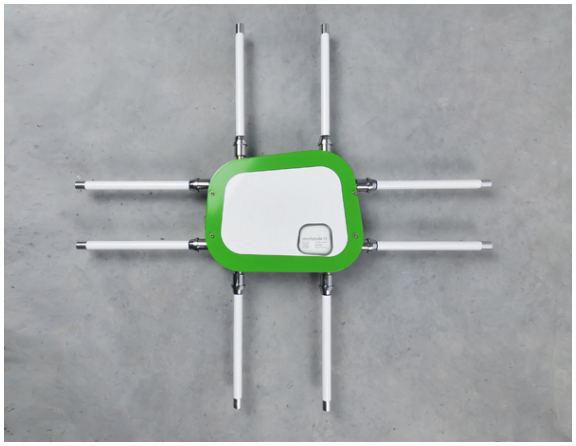

Figura C.1: Saxnet Meshnode III. Fuente: http://www.taiko-net.ch

- CPU: AMD Geode LX x86, 500 MHz
- Dos memorias flash: 16GB (principal), 4GB (recuperación)
- RAM: 512 MB
- 4 interfaces inalámbricas Atheros AR5001X+ (802.11a/b/g)
- Cada interfaz inalámbrica dispone de dos antenas

- Interfaz Ethernet: RTL-8169 Gigabit Ethernet
- Un puerto serie
- Distribución Linux Debian con kernel 2.6.24 por defecto
- Carcasa resistente al agua

Los nodos tienen dos modos de funcionamiento:

- *Normal system*: modo de funcionamiento normal en el que arranca en la memoria de flash de 16GB, en la que están instaladas la distribución incluida por defecto y la instalación personalizada que se ha realizado.
- *Rescue system*: modo de funcionamiento de recuperación que arranca en la tarjeta de 4GB, que tiene instalada una distribución reducida para reparar algún problema que ocurra con la memoria principal.

Para cambiar entre los dos modos, en la fuente tiene un interruptor para seleccionar el modo de funcionamiento. Hay que añadir que en modo funcionamiento normal la memoria hda es la memoria de 16GB y hdb es la memoria de 4GB. Sin embargo en modo de recuperación la memoria de 16GB es la hdb y la memoria de 4GB, la hda.

C.3. Instalación

C.3.1. Formateo e instalación de Debian

Para formatear la memoria principal, hubo que arrancar en modo *rescue system*, para no formatear la memoria que estaría siendo usada. Primero se chequeó la memoria principal para que no hubiera ningún error, luego se redimensionó el tamaño de la partición y por último con el comando `fdisk` se eliminó la partición que había y se crearon las dos particiones. Para realizar esos pasos se ejecutaron los siguientes comandos:

```
e2fsck -f /dev/hdb1
resize2fs /dev/hdb1 12G
fdisk /dev/hdb
delete
new -> p -> 1 -> 1-12601
new -> p -> 2 -> default
```

Después se inicia en modo normal (*normal system*), se le da formato a la nueva partición y se monta en el sistema con estos comandos:

```
mkfs -t ext2 /dev/hda2
mount /dev/hda2 /home/
```

El siguiente paso fue instalar la distribución Debian Squeeze 6. Se hizo con el comando `debootstrap`, que permite instalar una distribución basada en Debian desde otra distribución en cualquier directorio del sistema sin necesidad de CD:

```
wget http://ftp.es.debian.org/debian/pool/main/d/debootstrap/
debootstrap_1.0.26+squeeze1_all.deb
debootstrap -arch=i386 squeeze /home/ http://ftp.es.debian.org/debian/
```

Después de tener instalada la distribución, se instaló un kernel compilado con las mismas opciones que el que traían los nodos por defecto. Para instalarlo, se utilizó la herramienta `chroot`, que cambia el directorio de root por la ruta especificada. Antes de instalar el kernel fue necesario la instalación del paquete initramfs-tools para que creara todos los archivos necesario para arrancar en la carpeta `/boot`:

```
chroot /home/
apt-get install initramfs-tools
dpkg -i linux-image-2.6.39.2-mesh_2.6.39.2-mesh-10.00.Custom_i386.deb
dpkg -i linux-headers-2.6.39.2-mesh_2.6.39.2-mesh-10.00.Custom_i386.deb
```

Además se crearon los archivos necesarios para arrancar correctamente (`/etc/fstab`, `/etc/network/interfaces`) y se añadió la contraseña de root, ya que no estaba configurada:

```
# Fichero /etc/fstab
/dev/hda2 / ext2 defaults 0 1
proc /proc proc defaults 0 0

# Fichero /etc/network/interfaces
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp
passwd
```

Por último se modificó el grub para seleccionar la nueva instalación en el arranque:

```
title uc3m_meshnode
root (hd0,1)
kernel /boot/vmlinuz-2.6.39.2-mesh root=/dev/hda2 console=ttyS0,38400n8
initrd /boot/initrd.img-2.6.39.2-mesh
```

C.4. Configuración

C.4.1. Instalación drivers de las tarjetas inalámbricas

Se instalaron dos drivers Madwifi y ath5k para las tarjetas inalámbricas. Aunque se descartó el uso de Madwifi ya que era inestable, se explica su configuración.

C.4.1.1. Madwifi

Primero se instalaron los paquetes necesario para la instalación:

```
aptitude install module-assistant wireless-tools
```

El siguiente paso fue descargar las fuentes para instalar y descomprimirlas

```
wget http://snapshots.madwifi-project.org/madwifi-0.9.4-current.tar.gz
tar xzvf madwifi-0.9.4-current.tar.gz
```

Al intentar compilar se obtenían ciertos errores. Para evitarlos e instalar el driver se hicieron las siguientes modificaciones:

- Dentro de las fuentes, modificar la línea `EXTRA_CFLAGS+=`$(INCS)`(COPTS)` por `EXTRA_CFLAGS+=`$(INCS)`(COPTS) -Wno-error` en el fichero `net80211/Makefile` .
- Dentro de la carpeta `/usr/src/linux-headers-2.6.39.2-mesh` ejecutar el comando `make scripts`.

Finalmente, se compila y se instala el driver indicándole la ruta del kernel:

```
make KERNELPATH=/usr/src/linux-headers-2.6.39.2-mesh/
make install KERNELPATH=/usr/src/linux-headers-2.6.39.2-mesh/
```

Después de realizar todos estos pasos ya se puede cargar el módulo del driver:

```
modprobe ath_pci
```

C.4.1.2. ath5k

Para disponer del módulo de ath5k se tiene que añadir al fichero de configuración del kernel, siguiendo la siguiente ruta al ejecutar `make menuconfig`:

```
Networking support --> Wireless -->
      <M> cfg80211 - wireless configuration API
      <M> Generic IEEE 802.11 Networking Stack (mac80211)
Device Drivers --> [*] Network device support --> Wireless LAN -->
      <M> Atheros Wireless Cards -->
      <M> Atheros 5xxx wireless cards support
```

Aún así fue necesario aplicar un parche[1] para configurar correctamente la interfaz inalámbrica y disponer de los canales permitidos en España en las dos bandas, ya que por defecto al cambiar de región siempre estaban disponibles los canales de Estados Unidos.

[1] https://dev.openwrt.org/browser/trunk/package/mac80211/patches/403-ath_regd_optional.patch

Además hubo que instalar la herramienta CRDA, que permite cambiar entre diferentes regiones y configurar la interfaz inalámbrica de forma adecuada según la regulación de cada país. Para instalarla se siguieron los siguientes pasos:

```
apt-get install python-m2crypto libgcrypt11 libgcrypt11-dev libiw-dev
wget http://wireless.kernel.org/download/wireless-regdb/wireless-
regdb-2011.04.28.tar.bz2
tar xvjf wireless-regdb-2011.04.28.tar.bz2
wget http://wireless.kernel.org/download/crda/crda-1.1.2.tar.bz2
tar xvjf crda-1.1.2.tar.bz2
cd wireless-regdb-2011.04.28.tar.bz2
mkdir /usr/lib/crda
cp regulatory.bin /usr/lib/crda/
cd ../crda
make
make install
```

También se instaló el comando `iw`, que permite entre otras opciones crear interfaces a partir de la tarjeta física, seleccionar la antena por la que se emite, modificar la potencia o la velocidad de transmisión, seleccionar el canal, etc. Para su instalación se siguieron los siguientes comandos:

```
apt-get install pkg-config libnl1 libnl-dev
wget http://linuxwireless.org/download/iw/iw-latest.tar.bz2
tar -xvf iw-latest.tar.bz2
cd iw-3.2/
make
make install
```

Para crear puntos de accesos con ath5k se necesita la aplicación `hostapd`. Fue necesario la instalación de la versión 0.7.3, ya que disponía de la opción de crear dos puntos de accesos en la misma interfaz inalámbrica. Para instalar `hostapd` se siguieron los siguientes pasos:

```
apt-get install libssl-dev
wget http://hostap.epitest.fi/releases/hostapd-0.7.3.tar.gz
tar xzvf hostapd-0.7.3.tar.gz
cd hostapd-0.7.3/hostapd
cp defconfig .config
```

En el fichero .config descomentar las líneas que se muestran a continuación, para que el programa funcione con los dos drivers.

```
CONFIG_DRIVER_NL80211=y
CONFIG_DRIVER_MADWIFI=y.
```

Además hay que añadir la siguiente línea para que encuentre las fuentes de Madwifi:

```
CFLAGS += -I../../madwifi-0.9.4-r4176-20111123/.
```

Se compila e instala el programa:

```
make
make install
```

Finalmente, para crear un punto de acceso, se necesita crear un fichero con la configuración. En las siguientes líneas se muestra el fichero de configuración para crear los dos puntos de acceso sobre la misma interfaz:

```
# Fichero de configuración de hostapd
driver=nl80211
hw_mode=g
channel=5
interface=ath1
ssid=NETC
bss=ath0
ssid=NETC-OPEN
```

C.4.2. Ciclos de escritura sobre la memoria

Los nodos incluyen como memoria principal una memoria flash, que tiene un límite de ciclos de escritura y no se puede cambiar. Para reducir estos ciclos las carpetas `/var` y `/tmp` se montan en un pendrive USB y la memoria principal se monta como sólo lectura.

Para montar la carpetas `/var` y `/tmp` en el USB fue necesario realizar dos particiones. Para ello se utilizó la herramienta GParted. Además se realizó una tercera partición por si se necesitaba espacio adicional.

Por último se modificó el fichero `/etc/fstab` de la siguiente manera para que aplicar esta configuración al arrancar:

```
/dev/hda2 / ext2 ro,defaults 0 1
proc /proc proc defaults 0 0
/dev/sda1 /tmp ext2 defaults,rw,exec 0 0
/dev/sda2 /var ext2 defaults,rw,exec 0 0
```

Si se deseaba cambiar el modo de la memoria principal a modo Lectura y escritura, sólo hay que ejecutar el comando:

```
mount -o remount,rw /
```

C.4.3. Enrutamiento

Los nodos tienen asignada una dirección estática privada. Para configurar la dirección IP y la ruta por defecto de los nodos, se configuró el archivo `/etc/network/interfaces` de la siguiente manera:

```
auto eth0
iface eth0 inet static
address 10.0.Y.1XX
netmask 255.255.255.0
network 10.0.Y.0
broadcast 10.0.Y.255
gateway 10.0.Y.X
```

Donde Y tiene los valores 139 o 140 según si se encuentra en la red 163.117.139.0/24 o en la red 163.117.140.0/24, respectivamente. Y XX tiene el valor del nodo correspondiente.

C.4.4. OpenVPN

Para crear la VPN, el primer paso fue crear las rutas para alcanzar la dirección IP en la que se encuentra la VPN, donde GW es el gateway configurado según el fichero anterior:

```
/sbin/ip route add 10.0.1.1/32 via GW dev eth0
/sbin/ip route add 10.0.2.1/32 via GW dev eth0
```

El siguiente paso fue copiar los certificados que se había generado en el servidor en los nodos y configurar los archivos `client_auth.conf` y `client_no_auth.conf` con las opciones necesarias y se arranca el programa OpenVPN:

```
/etc/init.d/openvpn start
```

Después de unos segundos se crea la interfaz del túnel, `tap`. El último paso es puentear la interfaz que se ha creado con la interfaz inalámbrica, para que el punto de acceso actúe de *bridge*, con el comando `brctl` de la siguiente manera:

```
brctl addbr br0
brctl addif br0 ath0
brctl addif br0 tap0
/sbin/ip addr add 10.0.1.2/24 dev br0
/sbin/ifconfig br0 up
```

Con toda esta configuración se tienen configurados los nodos inalámbricos de forma que están directamente conectados al programa Chilli, y todo el tráfico que reciban por la interfaz inalámbrica se enviarán al portal cautivo, que va a ser quien lo gestione.

Apéndice D

Instalación y configuración de las soluciones de movilidad

D.1. Introducción

En este apéndice se va a explicar como se han instalado las diferentes soluciones de movilidad y sus ficheros de configuración.

La instalación y configuración de nivel de enlace se ha explicado en el capítulo 5 y con más detalle en los anexos B y C.

D.2. Soluciones de nivel de red

Para ejecutar las soluciones de nivel de red se tuvo que compilar un nuevo kernel con las opciones de las que hacen uso las implementaciones. Se compiló un kernel para el servidor, que actúa como agente local en MIPv6 y como LMA en PMIPv6; para los nodos inalámbricos, que son MAGs en PMIPv6 (en MIPv6 no hace ningún kernel especial); y por último para el cliente, que actúa como nodo móvil en MIPv6.

D.2.1. Compilar kernel

En todos los equipos anteriores se instaló el kernel 2.6.39.2. Se compiló el kernel con los siguientes comandos:

```
wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.39.2.tar.gz
tar zxf linux-2.6.39.2.tar.gz
cd linux-2.6.39.2/
make oldconfig
make menuconfig
```

Se seleccionaron las siguientes opciones:

```
General setup
```

```
-> Prompt for development and/or incomplete code/drivers
[CONFIG_EXPERIMENTAL]
-> System V IPC [CONFIG_SYSVIPC]

Networking support [CONFIG_NET]
-> Networking options
-> Transformation user configuration interface [CONFIG_XFRM_USER]
-> Transformation sub policy support [CONFIG_XFRM_SUB_POLICY]
-> Transformation migrate database [CONFIG_XFRM_MIGRATE]
-> PF_KEY sockets [CONFIG_NET_KEY]
-> PF_KEY MIGRATE [CONFIG_NET_KEY_MIGRATE]
-> TCP/IP networking [CONFIG_INET]
-> The IPv6 protocol [CONFIG_IPV6]
-> IPv6: AH transformation [CONFIG_INET6_AH]
-> IPv6: ESP transformation [CONFIG_INET6_ESP]
-> IPv6: IPComp transformation [CONFIG_INET6_IPCOMP]
-> IPv6: Mobility [CONFIG_IPV6_MIP6]
-> IPv6: IPsec transport mode [CONFIG_INET6_XFRM_MODE_TRANSPORT]
-> IPv6: IPsec tunnel mode [CONFIG_INET6_XFRM_MODE_TUNNEL]
-> IPv6: MIPv6 route optimization mode [CONFIG_INET6_XFRM_MODE_ROUTEOPTIMIZATION]
-> IPv6: IPv6-in-IPv6 tunnel [CONFIG_IPV6_TUNNEL]
-> IPv6: Multiple Routing Tables [CONFIG_IPV6_MULTIPLE_TABLES]
-> IPv6: source address based routing [CONFIG_IPV6_SUBTREES]

File systems
-> Pseudo filesystems
-> /proc file system support [CONFIG_PROC_FS]
```

Se compiló el kernel con el comando en los PCs:

```
make-kpkg -initrd kernel_image kernel_headers
```

Para compilar el kernel de los nodos se realizó en un PC de 64 bits, por lo que fue necesario hacer una compilación cruzada con el siguiente comando:

```
DEB_HOST_ARCH=i386 make-kpkg -arch i386 -cross-compile - -rootcmd fakeroot
-initrd -append-to-version -mesh-v_kernel_image kernel_headers
```

Al tenerlos compilados, se procedió a su instalación en cada uno de los equipos:

```
dpkg -i linux-image-2.6.39.2-mipv6_2.6.39.2-mipv6-10.00.Custom_amd64.deb
dpkg -i linux-headers-2.6.39.2-mipv6_2.6.39.2-mipv6-10.00.Custom_amd64.deb
```

D.2.2. Instalación y configuración de MIPv6

Para instalar la implantación de UMIP hubo que ejecutar los siguientes comandos en los PCs que iban a actuar como agente local y como nodo móvil. El primer paso fue instalar los paquete necesarios:

```
apt-get install autoconf automake bison flex libssl-dev indent ipsec-tools
radvd
```

Seguidamente se descargó el paquete para su compilación:

```
cd /usr/src/
git clone git://git.umip.org/umip.git
cd umip/
```

Finalmente se instaló con los comandos:

```
autoreconf -i
CPPFLAGS='-isystem /usr/src/linux/include/' ./configure -enable-vt
make
make install
```

La opción `-enable-vt` habilita un terminal virtual para poder visualizar la tabla Binding Cache o la lista de Binding Update en el agente local o el nodo móvil.

Al tener MIPv6 instalado, sólo falta crear los ficheros de configuración en el nodo móvil y en el agente local. En el agente local se utilizó el siguiente fichero de configuración:

```
# Sample UMIP configuration file for a MIPv6 Home Agent
NodeConfig HA;

# Set DebugLevel to 0 if you do not want debug messages
DebugLevel 10;

# Interface of the HA connected to the home link
Interface ‘‘dummy0’’;

# Binding information
BindingAclPolicy 2001:720:410:1014::2 allow;
DefaultBindingAclPolicy allow;

# Enable IPsec static keying
UseMnHaIPsec disabled;
KeyMngMobCapability disabled;
```

Además, como el agente local tiene que sustituir al nodo móvil en la red hogar (en este caso se encuentra en la interfaz dummy0), se ejecuta el programa `radvd` para que realice esa función. Se lanzó con el siguiente fichero de configuración:

```
interface dummy0
{
   AdvSendAdvert on;
   MaxRtrAdvInterval 3;
   MinRtrAdvInterval 1;
   AdvIntervalOpt on;
   AdvHomeAgentFlag on;
   AdvHomeAgentInfo on;
```

```
   HomeAgentLifetime 1800;
   HomeAgentPreference 10;
   # Home Agent address
   prefix 2001:720:410:1014::1/64
   {
      AdvRouterAddr on;
      AdvOnLink on;
      AdvAutonomous on;
   };
};
```

En el nodo móvil se creó el siguiente archivo de configuración:

```
# Sample UMIP configuration file for a MIPv6 Mobile Node
NodeConfig MN;

# Set DebugLevel to 0 if you do not want debug messages
DebugLevel 1;

# Enable the optimistic handovers
OptimisticHandoff disabled;

# Disable RO with other MNs (it is not compatible
# with IPsec Tunnel Payload)
DoRouteOptimizationMN disabled;

# The Binding Lifetime (in sec.)
MnMaxHaBindingLife 60;

# List here the interfaces that you will use
# on your mobile node. The available one with
# the smallest preference number will be used.


Interface ‘‘wlan0’’ {
   MnIfPreference 1;
}


MnHomeLink ‘‘wlan0’’ {
  HomeAgentAddress 2001:720:410:1014::1;
  HomeAddress 2001:720:410:1014::2/64;
}


# Disable IPsec static keying
UseMnHaIPsec disabled;
KeyMngMobCapability disabled;
```

Además se configuró el nodo móvil para que no realizara el mecanismo de DAD. Para ello se ejecutaron los siguientes comandos:

```
sysctl -w net.ipv6.conf.all.dad_transmits=0
sysctl -w net.ipv6.conf.default.dad_transmits=0
sysctl -w net.ipv6.conf.wlan0.dad_transmits=0
```

Finalmente fue necesario configurar los nodos inalámbricos para que anunciaran prefijos diferentes. También se utilizó `radvd` con el siguiente fichero de configuración:

```
interface ath1
{
   AdvSendAdvert on;
   MaxRtrAdvInterval 0.07;
   MinRtrAdvInterval 0.03;
   MinDelayBetweenRAs 0.03;
   AdvIntervalOpt on;
   AdvHomeAgentFlag off;
   # AR1 address
   prefix 2001:720:410:1012::1/64
   {
      AdvRouterAddr on;
      AdvOnLink on;
      AdvAutonomous on;
   };
};
```

Este fichero es igual en los dos nodos móviles salvo el campo `prefix`.

Con los ficheros configurados se ejecutan los siguientes comandos en cada equipo:

```
#### Servidor - Agente local
radvd -C /etc/radvd.conf
mip6d -c /usr/local/etc/mip6d.conf

#### Nodo móvil
mip6d -c /usr/local/etc/mip6d.conf

#### Nodos inalámbricos
radvd -C /etc/radvd.conf
```

D.2.3. Instalación y configuración de PMIPv6

La instalación de PMIPv6 se hizo en el PC que actúa como LMA y en los nodos inalámbricos que hace de MAGs. Al igual que en MIPv6 en estos equipos hubo que instalar un kernel compilado como se ha descrito anteriormente.

Para su instalación se siguieron los siguientes pasos. Primeramente en los MAGs es necesario instalar un servidor de SYSLOG. Para ello se instalan lo siguientes paquetes:

```
apt-get install socklog sysklogd
```

Posteriormente en el fichero `/etc/syslog.conf` se añade la línea `local7.info /var/log/pmip_syslog.log` y se crea el fichero `/var/log/pmip_syslog.log` con el comando: `touch /var/log/pmip_syslog.log`

Se edita el fichero `/etc/default/syslogd` añadiendo la línea `SYSLOGD="-r"`. Finalmente se reinicia el servicio con el comando: `/etc/init.d/sysklogd restart`

Este programa tiene la posibilidad de ejecutar un servidor RADIUS para autenticar a los nodos móviles que se conectan. En nuestro caso no se utilizó, y en su lugar se crearon los ficheros `/etc/match` con el siguiente contenido:

```
20010720041010180000000000000000 00000000000000000000000fb561167c
20010720041010130000000000000000 00000000000000000000000156d849414
```

Donde la primera parte es el prefijo que se la va a asignar al nodo móvil y por tanto el que se va a anunciar en los mensajes RA. Y los últimos bits de la segunda parte son la dirección MAC del nodo móvil.

Para configurar la compilación sin servidor Radius fue necesario realizar las siguientes acciones en los MAGs:

1. Eliminar los siguientes flags del fichero `PMIPv6_v0.4.1/pmipv6-daemon-umip-0.4/Makefile`

   ```
   -lfreeradius-client -lfreeradius-client -lfreeradius-client
   -lfreeradius-client -lfreeradius-client
   ```

2. Eliminar del fichero `PMIPv6_v0.4.1/pmipv6-daemon-umip-0.4/configure` la línea:

   ```
   #define HAVE_LIBFREERADIUS_CLIENT 1
   ```

Después se instala el programa tanto en los MAGs como en el LMA con los siguientes comandos:

```
wget http://www.openairinterface.org/openairfiles/documents//PMIPv6/
PMIPv6_V0.4.1/PMIPv6_v0.4.1.tar.bz2
tar -xjvf PMIPv6 _v0.4.1.tar.bz2
cd PMIPv6_v0.4.1/pmipv6-daemon-umip-0.4/
autoreconf -i
./configure
make
make install
```

Posteriormente se crearon los ficheros de configuración para los MAGs y el LMA. El fichero del MAG tiene la siguiente configuración:

```
NodeConfig MAG;

DebugLevel 10;
```

```
DoRouteOptimizationCN disabled;
DoRouteOptimizationMN disabled;
UseMnHaIPsec disabled;
KeyMngMobCapability disabled;

ProxyMipMag "MAG1 testbed n1"{
     RFC5213TimestampBasedApproachInUse enabled;
     RFC5213MobileNodeGeneratedTimestampInUse disabled;
     RFC5213FixedMAGLinkLocalAddressOnAllAccessLinks fe80::211:22ff:fe33:4455;
     RFC5213FixedMAGLinkLayerAddressOnAllAccessLinks 00:11:22:33:44:55;
     RFC5213EnableMAGLocalRouting enabled;
     LmaAddress 2001:720:410:1016::1;
     MagAddressIngress 2001:720:410:1013::1;
     MagAddressEgress 2001:410:140::6;
     MagDeviceIngress "ath1";
     MagDeviceEgress "eth0";
## Value in milliseconds
     PBULifeTime 40000;
## Value in milliseconds
     RetransmissionTimeOut 1000;
     MaxMessageRetransmissions 5;
     TunnelingEnabled enabled;
     DynamicTunnelingEnabled enabled;
     RadiusClientConfigFile ‘‘/usr/local/etc/radiusclient/radiusclient.conf’’;
     RadiusPassword ‘‘linux’’;
     PcapSyslogAssociationGrepString ‘‘A wireless client is associated - ’’;
     PcapSyslogDeAssociationGrepString ‘‘A wireless client is deauthenticated - ’’;
}
```

El fichero del LMA tiene la siguiente configuración:

```
# PMIPv6 LMA configuration file
NodeConfig HA;
# Set DebugLevel to 0 if you do not want debug message
DebugLevel 10;
LMAInterfaceMAG ‘‘eth1’’;

# IPsec configration - NO IPSEC AT THE MOMENT
UseMnHaIPsec disabled;
KeyMngMobCapability disabled;
```

Finalmente se lanzan los programas con los scripts que tiene creado el programa en la carpeta `PMIPv6 _v0.4.1/pmipv6-daemon-umip-0.4/extras`

Para el LMA:

```
./UMIP0.4_LMA_UBUNTU.10.04.py
```

Para los MAGs:

```
./UMIP0.4_MAG1_UBUNTU.10.04.py
```

Glosario

La lista de los acrónimos utilizados en este proyecto es la siguiente:

3GPP 3GPP 3rd Generation Partnership Project

ANDSF Access network discovery and selection function

AR Access Router

AP Access Point

BA Binding Acknowledgement

BCE Binding Cache Entry

BSS Basic Service Set

BU Binding Update

CN Corresponsal Node

CoA Care-of Address

CRDA Central Regulatory Domain Agent

DAD Duplicate Address Detection

DAR Dynamic Address Reconfiguration

DHCP Dynamic Host Configuration Protocol

DNS Domain Name System

ESS Extended Service Set

GGSN Gateway GPRS Support Node

GPRS General Packet Radio Service

GSM Groupe Spécial Mobile

GTP GPRS Tunnelling Protocol

HA Home Agent

HLR Home Location Register

HoA Home Address

ICMP Internet Control Message Protocol

IEEE Institute of Electrical and Electronics Engineers

IETF Internet Engineering Task Force

IP Internet Protocol

IQR Interquartile range

LMA Local Mobility Anchor

MAC Media Access Control

MAG Mobile Access Gateway

MD Movement detection

MIPv6 Mobile IPv6

MN Mobile Node

MN-ID Mobile Node Identifier

MPCTP Multipath TCP

mSCTP mobile SCTP

NAT Network Address Translation

NEMO Network Mobility

NETLMM Network-based Localized Mobility Management

NUD Neighbor Unreachability Detection

PBA Proxy Binding Acknowledgement

PBU Proxy Binding Update

PMIPv6 Proxy Mobile IPv6

QoS Quality of Service

RA Router Adverstisement

RAI Routing Area Identity

RS Router Solicitation

RTP Real Time Protocol

SCTP Stream Control Transmission Protocol

SDP Session Description Protocol

SGSN Serving GPRS Support Node

SIP Session Initiation Protocol

SSID Service Set IDentifier

TCP Transmission Control Protocol

UA User Agent

UAM Universal Access Method

UDP User Datagram Protocol

UE User Equipment

UMTS Universal Mobile Telecommunications System

VoIP Voice over Internet Protocol

VPN Virtual Private Network

XID eXchange Identification

Bibliografía

[20003] IEEE Std 802.11f 2003. IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11TM Operation. IEEE, 2003.

[802] IEEE Std 802.11-2007. IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control MAC and Physical Layer PHY Specifications. IEEE.

[80209] IEEE Std 802.21-2008. IEEE Standard for Local and Metropolitan Area Networks Part 21: Media Independent Handover Services. IEEE, January 2009.

[and] TS 24.312, Access Network Discovery and Selection Function (ANDSF) Management Object (MO). http://www.3gpp.org/ftp/Specs/html-info/24312.htm.

[Bat02] Regis J. Bates. *GPRS: general packet radio service.* McGraw-Hill, 2002.

[BFB+08] Ł. Budzisz, R. Ferrús, A. Brunstrom, K. Grinnemo, R Fracchia, G Galante, and F. Casadevall. Towards transport-layer mobility: Evolution of SCTP multihoming. *Computer Communications*, (Volume 31 Issue 5):980–998, March, 2008.

[CDMG06] A. Conta, S. Deering, and Ed. M. Gupta. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. RFC 4443, March 2006.

[chi] ChilliSpot. http://www.chillispot.info, última vez visitada Agosto 2012.

[CP02] Ed. C. Perkins. IP Mobility Support for IPv4. RFC 3220, January 2002.

[DH98] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, December 1998.

[DWPT05] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert. Network Mobility (NEMO) Basic Support Protocol. RFC 3963, January 2005.

[Edd04] W. M. Eddy. At what layer does mobility belong? *IEEE Communications Magazine*, 42 Issue 10:155–159, October 2004.

[GLD+08] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil. Proxy Mobile IPv6. RFC 5213, August 2008.

[GSW11] Mark Grayson, Kevin Shatzkamer, and Klaas Wierenga. *Building the Mobile Internet.* Cisco Press, 2011.

[HSK05] Ian Herwono, Joachim Sachs, and Ralf Keller. Performance Improvement of Media Point Network using the Inter Access Point Protocol according to IEEE 802.11f. *Wireless Conference 2005 - Next Generation Wireless and Mobile Communications and Services (European Wireless), 11th European*, pages 798–804, Apr, 2005.

[Jan09] Philipp K. Janert. *Gnuplot in Action: Understanding Data with Graphs*. Manning Publications, 2009.

[JK07a] Ed. J. Kempf. Goals for Network-Based Localized Mobility Management (NETLMM). RFC 4831, April 2007.

[JK07b] Ed. J. Kempf. Problem Statement for Network-Based Localized Mobility Management (NETLMM). RFC 4830, April 2007.

[KAL+05] Heikki Kaaranen, Ari Ahtiainen, Lauri Laitinen, Siamäk Naghian, and Valtteri Niemi. *UMTS networks: architecture, mobility and services*. John Wiley & Sons, 2005.

[KP07] Rajeev S. Koodli and Charles E. Perkins. *Mobile inter-networking with IPv6 : concepts, principles, and practices*. John Wiley & Sons, 2007.

[KR09] James F. Kurose and Keith W. Ross. *Computer Networking: A Top-Down Approach*. Addison Wesley, 2009.

[LJW11] M. Liebsch, S. Jeong, and Q. Wu. Proxy Mobile IPv6 (PMIPv6) Localized Routing Problem Statement. RFC 6279, June 2011. Errata Exist.

[MK04] J. Manner and M. Kojo. Mobility Related Terminology. RFC 3753, June 2004.

[NNSS07] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. Neighbor Discovery for IP version 6 (IPv6). RFC 4861, September 2007.

[ope] OpenVPN. http://www.openvpn.net, última vez visitada Agosto 2012.

[pep] PepperSpot. http://pepperspot.sourceforge.net, última vez visitada Agosto 2012.

[Per98] Charles E. Perkins. *Mobile IP: design principles and practices*. Addison-Wesley, 1998.

[PJA04] C. Perkins, D. Johnson, and J. Arkko. Mobility Support in IPv6. RFC 3775, June 2004. Obsoleted by: 6275.

[PJA11] C. Perkins, D. Johnson, and J. Arkko. Mobility Support in IPv6. RFC 6275, July 2011. Errata Exist.

[pmi] OpenAirInterface Proxy Mobile IPv6 (OAI PMIPv6). http://www.openairinterface.org/openairinterface-proxy-mobile-ipv6-oai-pmipv6.

[RM02] José María Hernando Rábanos and Cayetano Lluch Mesquida. *GPRS: tecnología, servicios y negocios*. Telefónica Móviles España, 2002.

[RSC+02] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 2543, June 2002.

[SBCM10] Ignacio Soto, Carlos J. Bernardos, María Calderón, and Telemaco Melia. PMIPv6: A Network-Based Localized Mobility Management Solution. *The Internet Protocol Journal, Volume 13, No.3*, September 2010.

[SCEB08] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier. Hierarchical Mobile IPv6 (HMIPv6) Mobility Management. RFC 5380, October 2008.

[Sol04] Hesham Soliman. *Mobile IPv6: mobility in a wireless Internet*. Addison-Wesley, 2004.

[Ste07] R. Stewart. Stream Control Transmission Protocol. RFC 4960, September 2007. Errata Exist.

[SXM+00] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. Stream Control Transmission Protocol. RFC 2960, October 2000. Obsoleted by: 4960, updated by: 3309.

[SXT+07] R. Stewart, Q. Xie, M. Tuexen, S. Maruyama, and M. Kozuka. Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration. RFC 5061, September 2007.

[TNJ07] S. Thomson, T. Narten, and T. Jinmei. IPv6 Stateless Address Autoconfiguration. RFC 4862, September 2007.

[umi] UMIP. http://www.umip.org/, última vez visitada Agosto 2012.

[wir] Linux wireless. http://linuxwireless.org/, última vez visitada Agosto 2012.

[WS99] Elin Wedlund and Henning Schulzrinne. Mobility support using SIP. *WOWMOM '99 Proceedings of the 2nd ACM international workshop on Wireless mobile multimedia*, pages 76 – 82, 1999.

[XCZ+07] Gaogang XIE, Ji CHEN, Hongxia ZHENG, Jianhua YANG, and Yu ZHANG. Handover Latency of MIPv6 Implementation in Linux. *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, pages 1780 – 1785, 2007.