Universidad Rey Juan Carlos
Escuela Superior de Ciencias Experimentales y Tecnología

Proyecto Fin de Carrera

Computación ubicua

Diego Chaparro González
dchaparro@acm.org

Tutor: Pedro de las Heras Quirós
pheras@gsyc.escet.urjc.es

30 de Junio de 2003

Resumen

La tecnología de comunicaciones inalámbricas es cada día más importante y empieza a estar tan extendida que empezamos a olvidarnos de la conexión por cables que hasta ahora era fundamental en algunos dispositivos, como teléfonos móviles u ordenadores portátiles. Esta tecnología ha experimentado un gran avance en los últimos tiempos, y su desarrollo sigue en aumento.

Los avances en comunicaciones inalámbricas han llevado a la creación de un nuevo campo en la computación, denominado computación ubicua. También están aprovechándose los avances en el campo de los componentes electrónicos, que llevan a la reducción del tamaño de los dispositivos y al aumento de su potencia. El desarrollo de la tecnología radioeléctrica, así como la difusión y abaratamiento de los dispositivos utilizados. El desarrollo de los protocolos de movilidad de dispositivos entre redes y los avances en el campo de los nuevos materiales.

Este nuevo sector de la computación, denominado computación ubicua o pervasiva, pretende incorporar a los objetos de la vida cotidiana capacidad de cómputo, de comunicaciones inalámbricas y de interacción entre ellos para crear un nuevo modelo de la realidad en la que estos objetos interoperan entre ellos para facilitar la realización de las tareas a las personas.

Para poder investigar acerca de este campo de la computación ubicua, se ha realizado un estudio desde diversos enfoques teóricos y prácticos de los campos más importantes en los que se basa esta tecnología.

En primer lugar se han estudiado las bases de la tecnología inalámbrica, y se han hecho experimentos reales sobre esta tecnología utilizando diversos dispositivos: conexiones entre PDAs mediante 802.11 e infrarrojos, conexión inalámbrica entre ordenadores con dispositivos dongles, ... En segundo lugar, después de haber estudiado la tecnología inalámbrica, se ha estudiado el segundo de los campos clave para la computación ubicua, es el campo de los protocolos de movilidad. Se ha estudiado y se han montado maquetas de redes de dispositivos inalámbricos para estudiar el funcionamiento de varias implementaciones de estos protocolos, se han hecho pruebas con estas maquetas realizadas y se han hecho experimentos para medir el rendimiento de los mismos.

Y por último, aplicando los dos campos estudiados anteriormente, junto con otros aspectos de las tecnologías de la comunicaciones, como redes ad-hoc, se han montado simulaciones de algunos escenarios reales que propone la computación ubicua utilizando las maquetas montadas en los capítulos anteriores adaptadas a estas situaciones, se ha implementado un protocolo de redes ad-hoc para los robots legos Mindstorm para comprobar el funcionamiento real de estas redes y se ha realizado una comparación de protocolos de movilidad sobre diferentes protocolos de red (IPv4 vs IPv6) entre otros experimentos.

Índice general

Índice de cuadros

Índice de figuras

Capítulo 1

Introducción

En este capítulo se presenta una introducción al contenido del proyecto. Se introducen las bases del estudio que se va a realizar en los siguientes capítulos, los objetivos que se pretenden conseguir en este estudio y la forma en la que se ha realizado.

1.1. Introducción

Las redes inalámbricas son una realidad hoy en día, estamos acostumbrados a ver ordenadores portátiles conectados a Internet sin necesidad de cables, pequeños ordenadores de mano conectados con los ordenadores de la oficina, cada día aumenta más la creación de redes inalámbricas ciudadanas, en la que voluntariamente y sin buscar beneficio más allá del uso de las tecnologías disponibles y el afán de aprender y practicar con ellas, hay ciudadanos que van poniendo a disposición de los demás puntos de acceso a una red que cada día va creciendo más, y que cada voluntario ayuda a que ésta crezca.

Y todo esto que cada día estamos y estaremos más acostumbrados a ver no es más que el inicio de un mundo de posibilidades que se abren con este nuevo modelo de computación, denominado computación pervasiva o ubicua. Este modelo de computación ubicua significa básicamente la omnipresencia de computadores muy pequeños interconectados sin cables que se incorporan de forma casi invisible a cualquier objeto de uso cotidiano, y usando pequeños sensores unidos a estos computadores pueden detectar el entorno que les rodea y tienen capacidades tanto de procesar información como de comunicación.

A partir de este modelo de computación son muchas las posibilidades que se pueden aprovechar, ya que estos dispositivos pueden no solo computar información y comunicarse con los demás sino que pueden detectar el entorno mediante diversos tipos de sensores, lo que les proporciona una interactividad continua con el entorno y les proporciona la capacidad de poder adaptarse a la diversas situaciones del entorno e incluso a cooperar con el resto de dispositivos disponibles en ese entorno para simular comportamientos casi "inteligentes".

Todas las posibles aplicaciones de estas tecnologías pueden verse aplicadas a la realidad gracias a los avances en diversos campos:

- La computación
- La microelectrónica
- La tecnología de la comunicación
- La ciencia de los materiales
- ...

Pero uno de los avances que más ha contribuido a ello son los avances en **microelectrónica**, que permiten que la capacidad de los micro-chips crezca de forma exponencial desde hace mucho tiempo y la tendencia continúa, como pronosticaba la *Ley de Moore*, por eso es normal que la capacidad de procesamiento de estos micro-chips se vaya multiplicando cada cierto tiempo y eso hace que cada día tengamos mayor capacidad de procesamiento por centímetro cuadrado de micro-chip. Al igual que sucede con la capacidad de procesamiento, también ocurre con otros factores de los dispositivos electrónicos, como la capacidad de almacenamiento, el ancho de banda de las comunicaciones y otros factores, que avanzan a un ritmo similar que dicha capacidad, con lo que conseguimos reducir cada día más el tamaño de los dispositivos con una gran capacidad de procesamiento, almacenamiento, ancho de banda y memoria sin aumentar el precio de los mismos.

Además, aparte de estos dispositivos de procesamiento cada vez más pequeños también se van desarrollando otro tipo de dispositivos que ayudan a aumentar las posibilidades en este campo, como son los micro-sensores que permiten recibir información, procesarla y devolver una respuesta con tan solo unos milímetros cuadrados de tamaño.

También se han realizado avances significativos en el campo de las **comunicaciones sin cables**, cuyos logros que más interesan en el modelo de computación ubicua son los de comunicaciones de corta distancia y con un bajo consumo de energía. Algunos de estos logros conseguidos es la tecnología WLAN, que permite la creación de redes locales con un alcance aproximado de unos 100-200 metros y con un ancho de banda de unos 10 Mb/s. También es importante la creación de redes de área personal (PAN), también llamadas redes de habitación sin cables, de las que el protocolo más importante es Bluetooth que permite un alcance de unos 10 metros con un ancho de banda aproximado de 1 Mb/s. Y la tecnología por infrarrojos, con el estándar Irda, que puede ser usado para recibir información sobre el entorno mediante sensores que perciben información sobre el mismo.

Con toda esta tecnología ya casi tenemos el entorno que pretende el modelo de **computación ubicua**, pero falta algo. Según el modelo del que disponemos actualmente para realizar comunicaciones entre dispositivos (protocolo TCP/IP), todo dispositivo está conectado a la red desde una localización determinada y se comunica con los demás mediante un identificador (dirección IP) que representa su situación actual, y si un dispositivo cambia de localización geográfica porque es un dispositivo móvil y tiene la capacidad para ello, entonces debe adquirir un identificador nuevo para comunicarse con los demás. Entonces, si no aplicamos nada durante este proceso, un dispositivo no puede ser localizado mediante un solo identificador si se mueve de un sitio a otro, porque este identificador cambia.

Y en este punto es donde entra en juego otra tecnología muy importante para que este modelo de computación siga adelante, y es la tecnología que proporciona la **movilidad** para dispositivos con los protocolos de micro y macro movilidad. Este tipo de protocolos permite que un dispositivo pueda utilizar su capacidad de movilidad entre redes sin que esto sea percibido por los demás dispositivos. Cualquier dispositivo podrá comunicarse con cualquier otro mediante los identificadores (direcciones IP) originales de los mismos independientemente del lugar de conexión a la red que tenga cada dispositivo en cada momento, y siempre podrán continuar con las comunicaciones en curso mientras se mueven de un sitio a otro. El protocolo más utilizado actualmente para realizar esta tarea es un protocolo estandarizado por el IETF llamado Mobile IP[Sol], que comúnmente es usado de forma conjunta con otros protocolos de micro-movilidad como por ejemplo Cellular IP.

1.2. Objetivos

En primer lugar, uno de los objetivos del presente trabajo es el de realizar un estudio sobre diferentes tecnologías que actualmente se están desarrollando a gran velocidad, y que en un futuro posiblemente cercano pueden hacer que cambie el modo de percibir el mundo de la computación actual que se basa en ordenadores personales conectados entre sí, a otro modo en el que las personas dejarán de percibir este modo de computación porque estos pequeños computadores estarán presentes en la mayoría de objetos cotidianos y pasarán desapercibidos.

Estos objetivos están escalonados. Primero se pretende realizar un estudio sobre la tecnología inalámbrica, estudiaremos las bases de esta tecnología y sus características más importantes. Una vez que hayamos hecho esto, el siguiente paso sería hacer experimentos sobre esta tecnología, debemos hacer pruebas con diversos dispositivos: PDA, ordenadores portátiles y de sobremesa, dispositivos de comunicación inalámbrica (dongles, beacons, ...), y otros dispositivos de comunicación inalámbrica poco frecuentes como los robots de Lego Mindstorms. Y realizar pruebas de conexiones entre estos dispositivos sobre las diferentes tecnologías inalámbricas estudiadas anteriormente, como infrarrojos y 802.11. De esta forma podremos observar la interacción entre diferentes dispositivos sobre diversos medios de comunicación y comprobar su rendimiento.

Una vez que hemos realizado estos experimentos, ésto nos servirá de base para la realización del estudio y experimentos sobre el siguiente campo de interés: la movilidad entre redes. Después de realizar un estudio de los diferentes protocolos de movilidad entre redes existentes, seleccionaremos varias implementaciones de estos protocolos (Mobile IP, Cellular IP, ...) y las llevaremos a la práctica. Crearemos maquetas de redes en las que podamos instalar estas implementaciones, y probar su funcionamiento, comprobar su rendimiento e incluso probar la interacción entre varios de estos protocolos.

Después de esto, debemos agrupar todo el estudio y los experimentos realizados en los apartados anteriores para realizar el estudio y los experimentos sobre el principal campo de interés, que se basa en los anteriores, el campo de la computación ubicua. Utilizando las maquetas creadas y utilizadas para los experimentos en los apartados anteriores, deberemos adaptarlas para la simulación de escenarios reales del modelo propuesto por la computación ubicua: debemos mezclar los dispositivos y las redes inalámbricas, la posibilidad de movilidad que nos ofrece lo estudiado en segundo lugar y aplicarlo a situaciones reales. De esta forma, podremos comprobar si el modelo propuesto de computación ubicua hoy en día es lo suficientemente maduro para su uso, y si disponemos de la tecnología necesaria para llevarlo a cabo.

Y por último, como no podía ser de otro modo, el objetivo fundamental es obtener conclusiones sobre el modelo estudiado, observar su desarrollo, el estado actual y el posible desarrollo futuro.

1.3. Organización del proyecto

En este **primer capítulo** se presenta una introducción a la temática general sobre la que versarán las distintas partes del proyecto, así como una lista inicial de los objetivos que se pretenden cubrir. También se presentan otros aspectos relacionados con la realización del trabajo, como herramientas y lenguajes utilizados tanto para la realización de la documentación como de los ensayos prácticos relacionados con la temática del proyecto.

En el **segundo capítulo** se muestran las bases sobre las que se sustenta la tecnología inalámbrica, presentando sus aspectos más importantes y la diversidad de este tipo de tecnologías. Después se describen algunos de los protocolos más utilizados en este campo, y se definen algunos de los experimentos realizados sobre él, usando diversos dispositivos y sobre diferentes protocolos de comuncación inalámbrica.

En el **tercer capítulo** se describen las tecnologías que permiten dotar a los dispositivos de la movilidad necesaria para el campo de la computación ubicua, protocolos de macro/micro movilidad. Y se describen los experimentos y prácticas realizadas sobre estos protocolos mediante el montaje de maquetas de redes en las que poder probar estas protocolos.

En el **cuarto capítulo** se desarrolla el modelo de computación ubicua en base a los campos de investigación descritos en los capítulos anteriores. Se describen algunos escenarios propuestos por este modelo de computación, y se experimenta con alguno de estos escenarios llevándolos a la práctica real.

Y por último, en el **quinto capítulo** se presentan las conclusiones extraídas de los estudios y de los experimentos realizados durante el desarrollo del proyecto.

1.4. Lenguajes, herramientas y tecnología

Como las bases sobre las que se sustenta el trabajo forman parte de los últimos avances en diversas tecnologías, para el desarrollo del mismo así como para los diversos ensayos prácticos realizados se ha decidido la utilización de sistemas que permitan un control total sobre el estado de los dispositivos utilizados, así como la posibilidad de acceder a las fuentes de todo el software utilizado, tanto para realizar estudios sobre él como para poder ser modificado o mejorado para adaptarlo a las necesidades del entorno, y todo esto encaja perfectamente con el modelo de software libre existente en la actualidad. Por ello casi la totalidad de los dispositivos utilizados (ordenadores personales, PDAs, portátiles, ...) utilizan un sistema operativo libre, como Debian GNU/Linux, y todo el software utilizado también está enmarcado dentro de esta categoría del software libre.

Ha sido necesario el conocimiento y estudio de diversa tecnología de programación para la puesta en práctica de los ensayos, programación en C, C++ sobre la que están desarrollados la mayoría de los protocolos de movilidad estudiados, programación de shell (GNU Bourne-Again Shell) y perl para la realización de scripts de automatización de pruebas. Así como programación sobre tcl para la mejora de cierto software de comunicaciones multimedia.

Por supuesto, también han sido necesarios amplios conocimientos sobre la administración de sistemas, administración de redes e instalación y configuración de dispositivos, ya que los ensayos han requerido la instalación de gran cantidad

de software de comunicaciones, diseño e implementación de redes adaptadas a dichos ensayos, instalación y configuración de dispositivos de comunicaciones no convencionales (como beacons, dongles, ...) y puesta en práctica de los ensayos diseñados.

Para todo ello se ha utilizado una gran cantidad de software, desde el software de los protocolos estudiados, herramientas de diseño, control y monitorización de redes, software multimedia para las pruebas, software para la creación de la documentación, ...

A continuación se puede ver un sumario de toda la tecnología utilizada:

- **Hardware** usado:
  - Compaq Ipaq
  - HP Jornada
  - Tarjetas inalámbricas 802.11b
  - Dongle
  - Beacon
  - Ordenadores portátiles y de sobremesa
  - Cámaras de videoconferencia
  - Material para creación de redes: hubs, cables de red, ...
  - Robots Legos Mindstorms
- **Sistemas operativos** usados:
  - Debian GNU/Linux
  - Familiar
  - Windows CE
  - LegOS
- **Lenguajes de programación** utilizados:
  - C
  - C++
  - perl
  - shell script
  - tcl
- **Tecnologías** usadas:
  - 802.11b
  - Irda
  - Mobile IP
  - Cellular IP
  - IPv6, IPv4
  - HTTP

1.5. La documentación

La documentación del proyecto se ha realizado utilizando el sistema de composición de textos TEX de Donald E. Knuth con la ayuda de los macros LATEX2e de Leslie Lamport, utilizando la versión teTEX versión 1.0 bajo Debian GNU/-Linux Sarge.

Para la edición del documento se han utilizado los editores vim versión 6.1 y xemacs versión 21.4.

Este documento se distribuye con las condiciones de la licencia **GFDL** [Fun02].

Capítulo 2

Tecnología inalámbrica

2.1. Historia de la tecnología inalámbrica

Los primeros inicios de la tecnología inalámbrica se produjeron sobre 1896, cuando *Gublielmo Marconi* inventó el telégrafo y en 1901 se produjo el primer envío sobre el Oceáno Atlántico. Este fue el primer momento en el que se utilizó una tecnología para poder enviar caracteres codificados mediante señales analógicas sin cables. A partir de entonces han surgido muchos dispositivos que utilizan este tipo de tecnología: radio, televisión, teléfono móvil, satélites de comunicaciones, ... Últimamente los avances y estudios más significativos están centrados en los satélites de comunicaciones, la tecnología celular y las redes inalámbricas.

En 1960 se lanzaron por primera vez satélites de comunicaciones, que en aquel entonces podían manejar un escaso volumen de tráfico, y hoy en día estos satélites son capaces de soportar el tráfico de comunicaciones de voz y de televisión entre países.

Las redes inalámbricas están permitiendo el desarrollo de redes **WAN**, **MAN** y **LAN** inalámbricas. En estas redes los protocolos más usados son 802.11 y Bluetooth

Los teléfonos móviles permiten comunicaciones duales entre dos extremos, como el telégrafo de Marconi. La primera generación de teléfonos móviles utilizaba tecnología analógica, con terminales muy pesados y con poco ancho de banda para las comunicaciones. Actualmente se utiliza tecnología digital, lo que permite mayor ancho de banda, mayor calidad de recepción y mayor seguridad.

Los estándares que definen la interacción entre dispositivos inalámbricos están avanzando muy rápidamente, lo que nos llevará de forma rápida a la creación de redes inalámbricas globales que permitirán acceso desde diferentes dispositivos con tecnología diferente y ofrecerán una amplia gama de servicios.

2.1.1. La revolución de la telefonía inalámbrica

Una de los mayores progresos en tecnología inalámbrica en los últimos tiempos, ha sido sin duda la revolución de los teléfonos móviles. En 1990 el número de usuarios de esta tecnología en todo el mundo era aproximadamente de 11 millones, y en 1996 el número de nuevos usuarios de teléfonos móviles superaba al número de nuevos usuarios de telefonía fija.

Las razones para justificar el éxito de esta tecnología son claras:

- Los teléfonos se mueven junto con las personas, y son independientes de la situación geográfica, por tanto permiten toda la movilidad que los usuarios necesitan.
- Los teléfonos cada vez son más pequeños, con mayores funcionalidades y tienen baterías más duraderas.
- Por otro lado, hay ciertos lugares apartados de centros urbanos en los que implantar servicios de tecnología fija por cable es muy costoso, mientras que la implantación de estaciones base de telefonía móvil es mucho más barato.

Además de esto, están apareciendo nuevos dispositivos de telefonía móvil con los que se puede tener acceso a Internet, aunque con una prestaciones muy bajas. Con estos dispositivos se pueden tener sistemas de mensajería instantánea, correo electrónico, y otras funcionalidades cada día más importantes.

2.1.2. Globalización de las redes de telefonía

Hoy en día no existe una única red de telefonía móvil, los dispositivos soportan una o dos de las tecnologías existentes y se conectan mediante un operador de telefonía. Para que esto no ocurra en el futuro y exista una compatibilidad entre la variedad de tecnologías es necesario la definición de estándares que regulen todo esto.

La primera generación de redes digitales inalámbricas aparecía en Norte América bajo el nombre de "Sistema de telefonía móvil avanzada" (AMPS: *Advanced Mobile Phone System*), que utilizaba un servicio de comunicaciones (CDPD: *Cellular Digital Packet Data*) que ofrecía un ancho de banda para comunicaciones de datos de 19.2 kbps. El CDPD utiliza los momentos de inactividad en las transmisiones por los canales de voz para ofrecer el servicio de comunicaciones de datos.

La segunda generación de sistemas inalámbricos se corresponde con el "Sistema Global de Comunicaciones Móviles" (GSM: *Global System for Mobile Communications*), el Servicio de Comunicaciones Personales (PCS: *Personal Communications Service*) IS-136 y el IS-95.

Como hemos dicho, es necesario que para las próximas generaciones de sistemas inalámbricos estén definidos los estándares que permitan el acceso global mediante los dispositivos inalámbricos. Para ello, la Unión Internacional de Telecomunicaciones (ITU: *International Telecommunication Union*) está desarrollando el IMT-2000 (textitInternational Mobile Telecommunications-2000), que es una familia de estándares, desarrollado en la banda de frecuencia de los 2 Ghz, cuya intención es proporcionar una red de comunicaciones inalámbrica global, definiendo las frecuencias de uso, los métodos de codificación y las transmisiones.

Además de esto, los estándares necesitan definir la interacción de los dispositivos inalámbricos con Internet. Para ello el WAP (*Wireless Application Protocol*) Forum está desarrollando un protocolo común que permita a los dispositivos con una pantalla y unos dispositivos de entrada limitados el acceso a Internet.

Y por último, el IETF (*Internet Engineering Task Force*) está desarrollando el estándar Mobile IP para adaptar el protocolo IP al nuevo entorno de dispositivos móviles.

2.1.3. El siguiente paso

El primer reto, por tanto, de la tecnología inalámbrica se ha centrado en las comunicaciones por voz, y como podemos observar hoy en día ha tenido un gran éxito y un desarrollo muy rápido.

El siguiente reto para esta tecnología consiste en las comunicaciones de datos, con la que se pretende que el acceso a Internet pueda realizarse al igual que hoy se realiza mediante las redes de cables. En realidad, no se pretende que el acceso sea el mismo, porque los dispositivos con los que se va a realizar ese acceso no son iguales: los dispositivos inalámbricos poseen pantallas de capacidad limitada y posibilidad de interacción con el usuario también más limitada que las disponibles mediante un ordenador personal.

2.2. Bases de la tecnología inalámbrica

En las siguientes secciones se presentan algunas de las bases sobre las que se sustenta el campo de las comunicaciones inalámbricas.

2.2.1. Transmisión de datos analógicos y digitales

Primero, algunas definiciones. Definimos **datos** como entidades que tienen significado o información. Las **señales** son eléctricas o electromagnéticas y son la representación de los datos. Y por último una **trasmisión** es el proceso de comunicación de datos mediante la propagación y procesamiento de las señales.

Datos digitales y analógicos

La transmisión de los datos se puede hacer de forma analógica o digital. Básicamente los **datos analógicos** son valores continuos en un intervalo de tiempo. Por ejemplo, el audio y el vídeo son datos analógicos, así como la mayoría de los datos que son recogidos por los sensores, como temperatura o presión. Por otra parte los **datos digitales** son aquellos que corresponden a valores discretos, como por ejemplo un texto de caracteres y números.

Señales analógicas y digitales

Una señal analógica es una variación electromagnética continua de onda que puede ser propagada sobre una diversidad de medios dependiendo de la frecuencia (cable coaxial, cable cruzado, el aire, ...) .

Una señal digital es una secuencia de pulsos de voltaje, que pueden ser representados mediante ceros y unos.

La principal ventaja de las señales digitales es que suele ser mucho más barata y es menos susceptible a interferencia por ruidos. Y la principal desventaja es que las señales digitales ofrecen mayor atenuación que las señales analógicas.

Tanto los tipos de datos analógicos como digitales pueden ser representados y propagados mediante señales analógicas o digitales:

- Los datos digitales pueden ser representados mediante señales digitales mediante el uso de un módem. Un módem convierte una serie de pulsos binarios en una señal analógica modulando la frecuencia. La mayoría de los módem tradicionales representan datos digitales en el espectro de voz y permiten que estos datos sean propagados sobre las lineas tradicionales de telefonía. En el otro extremo, otro módem demodula la señal para recuperar los datos originales.

- Los datos analógicos pueden ser representados por señales digitales mediante una operación muy similar. El dispositivo que realiza esta función para datos de voz se denomina codec. Esencialmente, lo que hace un codec es recoger una señal analógica que representa datos de voz, muestrea periódicamente esa onda y convierte su amplitud en una unidad numérica que es representada digitalmente. En el otro lado de la línea otro codec usa la señal digital para decodificarla en la señal analógica original.

Transmisiones analógicas y digitales

Tanto las señales analógicas como las digitales deben ser transmitidas mediante un medio adecuado, y la forma de tratarlas dependerá del sistema de transmisión.

- La transmisión analógica consiste en transmitir señales analógicas independiente del contenido (pueden ser datos digitales o analógicos). Pero en ambos casos la señal analógica sufrirá atenuaciones que limitan la distancia del transporte de dicha señal en el medio empleado. Para aumentar la distancia de transmisión se usan amplificadores que aumentan la energía de la señal, pero también aumenta la cantidad de ruido en la misma. En datos analógicos, como la voz, un poco de ruido en la transmisión es admisible, pero en los datos digitales esto no es válido.
- Sin embargo, la transmisión digital está relacionada con el contenido de la señal. Y la señal digital también posee un límite de distancia antes de que se pierda la integridad de los datos, por ello, para aumentar la distancia de transmisión, se usan repetidores. Un repetidor recibe una señal digital, vuelve a crear la cadena de patrones de unos y ceros, y vuelve a transmitir la nueva señal creada. Con esta solución, la atenuación en este tipo de transmisión está superada. Esta misma señal puede ser utilizada si una señal analógica contiene datos digitales. Se usan dispositivos para retransmitir la señal, los cuales reciben la señal analógica, recuperan los datos digitales y crean una nueva señal analógica sin ruido.

2.2.2. Espectro, medio de transmisión

Capacidad del canal

Una variedad de razones pueden ser las causas para distorsionar o atenuar una señal. Una de las más usuales suele ser la inclusión de **ruido** en la señal, por la que la señal se distorsiona. Para los datos digitales, lo que realmente nos interesa es: ¿cuánto de importante son estas limitaciones para la tasa de datos que se puede conseguir?.

La tasa máxima a la que los datos pueden ser transmitidos sobre un determinado canal de comunicación bajo determinadas circunstancias es la **capacidad del canal**.

Conceptos importantes:

- **Tasa de datos**: Es la tasa, en bits por segundo (bps) a los que pueden ser transmitidos los datos.
- **Ancho de banda**: Es el ancho de banda de la señal transmitida debido al emisor y a la naturaleza del medio de comunicación. Se expresa en ciclos por segundo (Hertz).
- **Ruido**: Es el nivel medio de ruido sobre el medio de comunicación.
- **Tasa de ruido**: Es la tasa en la que suceden los errores: se transmite un uno y se recibe un cero, y viceversa.

	Rango de frecuencia
Radio	30MHz - 1GHz
Micro-ondas	1GHz - 40GHz
Infrarrojos	$3x10^11$ - $2x10^14$ Hz

Cuadro 2.1: Espectro electromagnético para telecomunicaciones inalámbricas

El problema que nos encontramos es que las mejoras en las comunicaciones son costosas, a mayor ancho de banda disponible mayor es el coste asociado. Por eso, lo que buscamos es obtener la mayor tasa de datos posible con un determinada tasa de error para un ancho de banda dado. Y el principal problema para conseguir esto es el ruido.

Si tuviéramos un canal que estuviera libre de ruido, la limitación de la tasa de datos sería simplemente el ancho de banda de la señal.

Si las señales a ser transmitidas son binarias, cada elemento de la señal puede representar un bit, pero podemos tener señales que en cada elemento tengan más de dos niveles, por tanto cada elemento representará más de un bit. Y con este método, dado un ancho de banda, podemos aumentar la tasa de datos. Pero el número de niveles de cada elemento de la señal viene impuesto por el ruido y las características del medio de transmisión.

Con esto podemos observar que si aumentamos el ancho de banda, aumentamos la tasa de datos. Pero el problema es que si aumentamos la tasa de datos, los bits ocupan menos espacio en la señal, y por tanto hay más bits que pueden ser afectados por el ruido. Y con esto tenemos que con un determinado nivel de ruido, a mayor tasa de datos mayor es la tasa de errores.

Medio de transmisión

En un sistema de transmisión de datos, el medio de transmisión es el camino físico entre el emisor y el receptor. Este medio de transmisión puede ser guiado o no guiado. En el primer caso las ondas electromagnéticas son transportadas por un componente sólido, como un cable coaxial, fibra óptica, ... Y en el segundo caso, las transmisiones no guiadas se realizan por la atmósfera y el espacio exterior, estas son las **transmisiones inalámbricas**.

Las calidad de la transmisión depende de las características del medio y de las características de la señal. En el caso de los medios guiados, el factor más importante para determinar las limitaciones de la transmisión es el medio empleado. Sin embargo, en el caso de las transmisiones por medios inalámbricos lo más importante es el ancho de banda de la señal que produce la antena emisora. Esta señal suele ser omnidireccional a frecuencias bajas, y direccional a frecuencias más altas.

En la tabla 2.1 se muestra una clasificación de tecnologías inalámbricas en función del rango de frecuencias en el que operan.

2.3. Redes locales inalámbricas

En los últimos tiempos, las redes locales inalámbricas han ocupado un gran lugar dentro del mercado de las redes locales. Las empresas se han dado cuenta de que las **WLAN** (*Wireless Local Area Network*) son un elemento indispensable junto con sus redes cableadas, porque con ellas se pueden satisfacer las necesidades de movilidad, redes ad-hoc o accesibilidad en lugares difíciles de cablear [Sta01].

Según [PC95] hay 4 áreas principales de aplicación de las redes inalámbricas:

Extensión de LAN

Hay situaciones en las que realizar el cableado de algunos lugares es una tarea muy complicada, por razones como edificios con grandes áreas abiertas, edificios antiguos que no tienen la infraestructura necesaria o pequeñas oficinas en las que cablearlas no es económico.

Para estas situaciones las redes locales inalámbricas son la solución, pero normalmente estas WLAN son solamente una parte de la red local, siempre hay otra parte de la red que contiene elementos que están mejor cableados: servidores y algunas estaciones de trabajo.

Interconexión de edificios

La interconexión de edificios cercanos también es una buena aplicación de las redes locales inalámbricas, porque permiten enlazar varios edificios sin necesidad de tener cableado entre ellos mediante enlaces punto a punto inalámbricos.

Acceso nómada o ubicuo

La computación ubicua también es otro de los servicios que se pueden ofrecer mediante las redes locales inalámbricas. Con ella los usuarios pueden conectarse a la red local mediante sus dispositivos móviles (portátiles, PDAs, ...) y esto lo pueden hacer desde diversos lugares físicos gracias a la cobertura que proporcione la red inalámbrica.

Redes Ad-hoc

Las redes ad-hoc son redes que se generan entre dispositivos que se unen a esta red descentralizada sin necesidad de una infraestructura creada anteriormente. Suelen ser redes temporales que se crean para situaciones concretas. Por ejemplo, una reunión de trabajadores en un despacho en la que éstos disponen de dispositivos móviles que forman una red para intercambiar datos.

Las redes inalámbricas se clasifican dependiendo de las técnicas de transmisión que usan, y todas estas redes se pueden enmarcar en alguna de las siguientes categorías:

- **Redes de área local por infrarrojos**:

  Las comunicaciones inalámbricas mediante infrarrojos están extendidas en los hogares en los dispositivos de control remoto, como mandos a distancia por ejemplo.

Pero las comunicaciones por infrarrojos también pueden ser usadas para crear redes locales. Estas redes ofrecen algunas ventajas con respecto al resto de redes inalámbricas:

- El espectro de comunicaciones inalámbricas es ilimitado porque no está regulado, con lo que podemos conseguir altas tasas de datos.
- Los infrarrojos comparten algunas propiedades de la luz que son muy interesantes para crear redes locales, como la propiedad de reflejarse sobre algunos objetos que hace que los infrarrojos emitidos lleguen a más lugares.
- No penetran las paredes ni los objetos opacos, lo que ofrece ventajas para tener redes seguras dentro de espacios cerrados y posibilidad de crear diferentes redes en diferentes espacios sin problemas por las posibles interferencias.

Además, crear este tipo de redes es más económico que otras redes inalámbricas y más sencillo.

Por otro lado, estas redes también tienen desventajas, como el ruido que se puede producir en entornos con mucha luz o elementos luminosos, y que la potencia de transmisión está limitada ya que puede ser dañino para los ojos y aumenta el consumo de energía.

- **Redes de área local de espectro expandido**:

  La mayoría de redes locales inalámbricas usan técnicas de espectro expandido. Estas redes, excepto las que son muy pequeñas, suelen usar un esquema de varias celdas con diferentes frecuencias para evitar interferencias.

  Dentro de cada celda se puede usar un sistema de conexión punto a punto o tipo hub. En las conexiones tipo hub hay un elemento central que proporciona la conexión inalámbrica a los dispositivos y todas las comunicaciones se realizan entre este elemento central y estos dispositivos, no hay comunicación directa entre dispositivos entre sí. Por otro lado, las conexiones punto a punto se utilizan usualmente para crear redes ad-hoc.

  Las bandas de frecuencia dependen de cada país. En EEUU hay tres rangos de frecuencia que no están regulados: la banda de los 915 MHz, la banda de 2.4-2.4835 GHz y la banda de los 5.8GHz.

- **Redes de área local de banda estrecha**:

  La tecnología de microondas de banda estrecha es el uso de la banda de radiofrecuencia para enviar señales, con un ancho de banda muy estrecho, lo justo para poder enviar la señal.

  Hasta ahora este tipo de tecnología operaba en frecuencias licenciadas, es decir, no liberadas para uso común. Pero recientemente han aparecido dispositivos que operan en frecuencias liberadas. En 1995 apareció RadioLAN, que opera en el espectro liberado ISM (*Industrial, Scientific, Medical*), y opera a baja energía (0.5 watios) y en la banda de los 5.8 GHz, con un alcance de unos 50-100 metros.

En las siguientes secciones se presentan los protocolos más usados actualmente para la creación de redes inalámbricas.

2.4. 802.11

Las redes 802.11 son hoy en día una realidad, y su uso es muy frecuente al contrario de otro tipo de tecnologías inalámbricas de las que se espera un gran futuro pero que todavía no son usadas comúnmente.

En las siguientes secciones se presentan las características y usos más comunes de estas redes, así como un ejemplo práctico de como configurar este tipo de dispositivos.

2.4.1. Arquitectura

Según el **modelo OSI** [Zim80] las capas superiores (nivel 3, 4 y superiores)) son independientes de la arquitectura de red, por tanto en este apartado para estudiar la arquitectura del protocolo solo nos interesan las capas inferiores.

En la figura 2.1 se muestra la arquitectura de protocolos para redes inalámbricas que ha sido adoptado para la realización de estándares con la tecnología 802.11. Se denomina **modelo de referencia IEEE 802.11**.

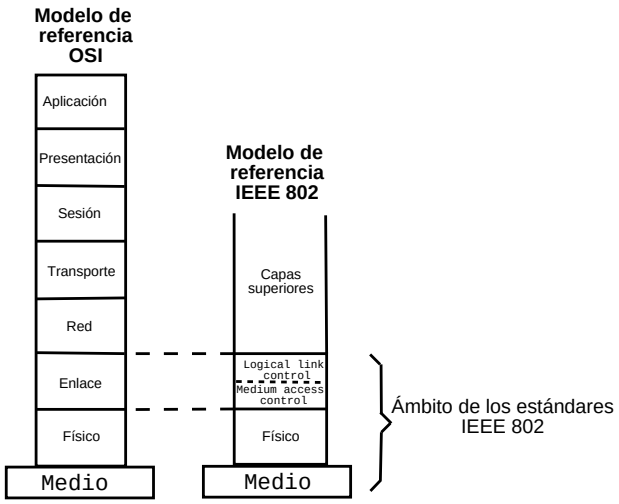


Figura 2.1: Capas de protocolos de IEEE 802 y modelo de referencia OSI

Las capa más baja del modelo de referencia 802 corresponde a la capa física del modelo OSI, y tiene la funcionalidad de esta última:

- Codificación/decodificación de señales
- Transmisión/recepción de bits

Pero además, el nivel físico del modelo 802 incluye la especificación para el medio de transmisión y la topología, considerado como funciones por debajo de las típicas del nivel físico del modelo OSI.

Por encima de la capa física está el nivel de enlace, que proporciona servicio a los clientes de la red inalámbrica. Las funciones de este nivel son:

- Cuando transmite, empaquetar los datos en una trama que incluye campos para la dirección y la detección de errores

Asociación
Autenticación
Des-autenticación
Disociación
Distribución
Integración
Entrega MSDU
Privacidad
Reasociación

Cuadro 2.2: Servicios de IEEE 802.11

- Cuando recibe, desempaquetar los datos y realizar la detección de errores.
- Controlar el acceso al medio de transmisión de la red local
- Proporcionar un interfaz a las capas superiores y realizar control de flujo y de errores.

En el modelo 802 los tres primeros se agrupan en un subnivel llamado Control de Acceso al Medio (MAC) y el último en un subnivel llamado Control de Enlace Lógico (LLC). Se puede ver el esquema en la figura 2.1.

Estos dos subniveles están diferenciados por dos razones. La primera es porque normalmente en el nivel de enlace no se encuentran las funciones para controlar el acceso al medio compartido. Y el segundo es que para el mismo LLC se pueden usar diferentes MAC, por eso deben estar separados.

2.4.2. Servicios

El IEEE 802.11 define nueve servicios que deben proporcionar las redes locales inalámbricas para prestar la misma funcionalidad que las redes locales cableadas. Estos nueve servicios son los siguientes:

- **Distribución**: Es el principal servicio usado por las estaciones para intercambiar el MAC cuando un paquete debe ser transferido a otra zona.
- **Integración**: Proporciona la transferencia de datos entre una estación de una red local 802.11 y otra de una red local 802.x
- **Entrega MSDU**: El MSDU (MAC Service Data Unit) es un bloque de datos que se entrega desde el Control de acceso al medio de usuario al nivel de control de acceso al medio.
- **Asociación**: Establece una relación inicial entre un punto de acceso (AP: Access Point) y una estación para que ésta pueda enviar o recibir frames de/desde la red WLAN.
- **Reasociación**: Habilita una asociación creada para ser transferida desde un punto de acceso a otro para permitir movilidad.
- **Disociación**: Notificación por parte del punto de acceso o de la estación para comunicar que la asociación ha terminado.

- **Autenticación**: Se usa para establecer la identidad de cada estación. Es necesario para que una estación pueda conectarse con un punto de acceso.
- **Des-autenticación**: Cuando la autenticación debe ser terminada.
- **Privacidad**: Para prevenir que el contenido de los mensajes transferidos pueda ser leído por otros que no sean el receptor. El algoritmo especificado en el estándar es WEP.

2.4.3. Especificaciones de 802.11

Existen varias especificaciones de 802.11, entre ellas las más usadas hoy en día son:

- **IEEE 802.11a**: Usa la banda de los 5 GHz, y puede alcanzar una tasa de datos de 54 Mbps.
- **IEEE 802.11b**: Funciona en la banda de los 2.4GHz y puede proporcionar una tasa de datos de hasta 11 Mbps.

2.4.4. Configuración de una tarjeta inalámbrica 802.11b

A continuación se va a presentar un ejemplo real de uso de la tecnología 802.11b, para ello se va a configurar y usar una tarjeta que proporciona comunicaciones a dispositivos usando este estándar.

La configuración de una tarjeta inalámbrica 802.11b, como con el resto de dispositivos, depende del modelo de la tarjeta, del sistema operativo usado y de la configuración del sistema. Pero como dijimos en el apartado 1.4, los sistemas usados para la realización de ensayos prácticos han sido equipos con sistema operativo GNU/Linux, en especial Debian GNU/Linux, por eso todas las configuraciones y prácticas que se expliquen se referirán a dicho sistemas.

Las tarjetas 802.11b utilizadas durante las pruebas han sido tarjetas PCMCIA Lucent Technologies y Compaq, pero todas ellas tenían chipset de Lucent. Para instalar una de estas tarjetas hay que seguir los siguientes pasos:

1. Instalar las fuentes del kernel de linux en /usr/src/linux sino están instaladas.
2. Instalar las fuentes del paquete PCMCIA en /usr/src/pcmcia
3. Configurar y recompilar las fuentes PCMCIA: usa configure y make en /usr/src/pcmcia.
4. Recompilar el kernel antes del paso anterior solo si es necesario.
5. Los ficheros de configuración se encuentran en /etc/pcmcia

Es aconsejable instalar las *wireless tools* para poder configurarla correctamente. Después de haberla instalado es necesario configurarla para adaptarla al entorno que necesitemos.

Configuración del nivel de enlace

Debemos definir el tipo de arquitectura de la red a la que pertenece la tarjeta. Existen varios modos, los más comunes son el modo infraestructura y modo *ad-hoc*, aunque últimamente también es muy común el poder poner estas tarjetas en modo *master*. En el modo infraestructura todos los dispositivos se conectan a un dispositivo central, normalmente un Punto de Acceso (AP), y todos estos dispositivos se comunican única y exclusivamente a través de este dispositivo. Sin embargo el modo ad-hoc se utiliza para que todos los dispositivos puedan establecer comunicaciones con cualquier otro, pueden hablar todos con todos. Esta último modo es aconsejable cuando no existen muchos nodos, o cuando las situaciones lo requieran: computación móvil, redes ad-hoc, ... Y el modo master sirve para que un dispositivo con una tarjeta configurada de esta forma pueda hacer de punto de acceso para que otros dispositivos se conecten a él.

Para cambiar el modo de una tarjeta se utilizan las aplicaciones que proporciona el paquete textitwireless tools. Si el interfaz de la tarjeta inalámbrica es eth0, y queremos cambiar el modo a modo infraestructura debemos ejecutar como superusuario:

```
> iwconfig eth0 mode managed
```

También se puede definir qué canal se quiere utilizar. La banda de frecuencia utilizada por estas tarjetas va desde los 2.4 GHz hasta los 2.4835, y este rango está dividido en 13 canales, que se pueden utilizar para no solapar redes y evitar interferencias. Por eso podemos elegir el canal en el que queremos que trabaje la tarjeta:

```
> iwconfig eth0 channel 10

\end{vebatim}

También se puede definir el ESSID, que no es más que un identificador
de la red a la que queremos conectarnos. Para configurar el ESSID:

\begin{verbatim}

> iwconfig eth0 essid nombre_red
```

Y también podemos definir el algoritmo de cifrado, la longitud de la clave, ...

Configuración del nivel IP

Aquí se debe configurar el nivel de red de la tarjeta. Esto puede ser algo trivial o algo complejo dependiendo del diseño de red que queramos montar. Para el caso más sencillo en el que la tarjeta se va a conectar a una red en la que existe un AP con el que debe de comunicarse, basta con asignarle los datos de red a la interfaz de la tarjeta. Por ejemplo:

```
> ifconfig eth0 192.168.240.10 netmask 255.255.255.255
> route add -net default gw 192.168.240.1
```

Claro, que toda esta configuración no es necesario hacerla cada vez que utilicemos la tarjeta. Toda esta configuración se puede almacenar en los ficheros de configuración del sistema para que no sea necesario repetir estas operaciones. En el fichero de configuración */etc/pcmcia/wireless.opts* se pueden configurar muchas de estas opciones.

2.5. Bluetooth

El objetivo de la tecnología Bluetooth es proporcionar una capacidad de comunicación universal de **corto alcance**. Funciona en la banda de los 2.4 GHz, como 802.11b, y puede alcanzar tasas de datos de hasta 720 kbps entre dos dispositivos a una distancia de unos 10 metros.

Bluetooth puede ofrecer a los usuarios servicios como:

- Utilizar unos cascos inalámbricos conectados a un teléfono mediante Bluetooth.
- Eliminar cables entre el ordenador y los periféricos como impresoras, teclados, ratones, ...
- Hacer una llamada telefónica a casa para activar servicios como alarmas o servicios de calefacción.

2.5.1. Aplicaciones

Bluetooth está diseñado para funcionar en entornos con muchos usuarios. Se pueden crear pequeñas redes de hasta 8 dispositivos llamadas **piconet**, y pueden coexistir 10 de estas piconets en el mismo espacio de cobertura de Bluetooth. Cada una de estas redes codifican sus datos y los protegen contra posibles intrusiones e interferencias.

Proporciona soporte para tres áreas de aplicación:

- **Puntos de acceso de voz y datos**: Proporciona transmisiones en tiempo real de voz y datos entre diferentes dispositivos.
- **Reemplazo de cables**: Elimina los cables, incluso propietarios, para conectar prácticamente cualquier tipo de dispositivo. La distancia máxima es de unos 10 metros, pero puede aumentarse hasta los 100 usando amplificadores.
- **Redes ad-hoc**: Dos dispositivos Bluetooth pueden establecer una conexión ad-hoc si se encuentran en el mismo rango de cobertura.

2.5.2. Estándares

Los estándares de Bluetooth son muy extensos, y están divididos en dos grupos:

- Las especificaciones del núcleo: describen los detalles de la arquitectura de capas del protocolo Bluetooth, desde el interfaz radio hasta el control de enlace
- Las especificaciones de perfiles: cada perfil describe el uso de la tecnología descrita en el grupo anterior de especificaciones para adaptarla a un modelo de uso concreto. Estos perfiles intentan especificar un estándar de interoperabilidad para que cualquier dispositivo pueda interactuar con cualquier otro.

Cada modelo de uso implementa una determinada aplicación basada en Bluetooth. Algunos de estos son:

- Transferencia de ficheros
- Puente a Internet: Un ordenador está conectado inalámbricamente a un teléfono o módem inalámbrico que le proporciona acceso a Internet, fax, ...
- Conexión a red de área local: Cuando los dispositivos están conectados a una piconet.
- Sincronización: sincronización de información entre diversos dispositivos, agenda de teléfono, calendario, ...
- Cascos de música inalámbricos.

Como hemos dicho antes, una piconet es una red de dispositivos inalámbricos que contiene un nodo maestro y hasta 7 dispositivos esclavos. El maestro selecciona el canal (secuencia de frecuencias de salto) y la fase de transmisión que deben utilizar todos los dispositivos de esa piconet.

Y una **scatternet** es cuando uno de los dispositivos pertenece a varias piconet, ya sea como esclavo o maestro de cualquiera de ellas. De esta forma, una gran cantidad de dispositivos pueden compartir el mismo espacio físico y aprovechar eficientemente el ancho de banda porque cada piconet tiene una frecuencia distinta del resto asignada.

2.6. IrDA

IrDA es un estándar definido por el IrDA Consorcio[Ass93], y especifica un tipo de comunicaciones inalámbricas por medio de radiaciones de infrarrojos. Es el protocolo por infrarrojos más extendido en la actualidad.

Las comunicaciones por infrarrojos ofrecen un servicio de bajo coste y fiable para conectar ordenadores y periféricos sin el uso de cables. Hoy en día todos los ordenadores portátiles poseen dispositivo de infrarrojos IrDA integrado, y también muchos dispositivos como teléfonos móviles o PDA.

IrDA presenta dos especificaciones básicas dentro de las cuales existen unos cuantos estándares para diversos dispositivos, que son *IrDA Data* e *IrDA Control*, y se describen a continuación.

2.6.1. IrDA Data

IrDA Data tiene como objetivo facilitar comunicaciones de alta velocidad a corto alcance, con visión directa y punto a punto para comunicaciones entre un ordenador y diversos dispositivos, como cámaras digitales, dispositivos de almacenamiento, ...

Consiste en una serie de protocolos obligatorios y algunos opcionales, que se describen a continuación:

Nivel físico

- Rango: Normalmente el rango máximo está entre uno y dos metros, pero hay modos especiales en los que si la distancia está entre los 20 o 30 centímetros se puede reducir el consumo hasta unas 10 veces menos.
- Comunicaciones bidireccionales
- Transmisión de datos desde los 9600 bps hasta los 115 kbps con una base de precio/coste, pero puede alcanzar una velocidad máxima de 4 Mbps
- Los paquetes de datos contienen código de detección de errores.

Protocolo de acceso de enlace (IrLAP)

- Proporciona comunicación fiable y ordenada entre dos dispositivos.
- Proporciona servicios de descubrimiento de dispositivos
- Maneja nodos ocultos

Protocolo de control de enlace (IrLMP)

- Proporciona multiplexación de la capa anterior (IrLAP). Pueden existir varios canales sobre una conexión IrLAP
- Proporciona un protocolo de descubrimiento de servicios.

Protocolos opcionales

Los protocolos anteriores son los básicos que debe tener toda comunicación de *IrDA Data*, pero existen otros protocolos específicos para diversos dispositivos o modos de operación:

- Tiny TP
- IrCOMM
- OBEX
- IrDA Lite
- IrTran-P
- IrMC
- IrLAN

2.6.2. IrDA Control

IrDA Control está diseñado para permitir comunicaciones entre periféricos como teclados, ratones o joystick y numerosos dispositivos como ordenadores, televisiones, consolas de videojuegos, ...

Al igual que *IrDA Data* posee una pila de protocolos base:

Nivel Físico

- Distancia de unos cinco metros
- Comunicaciones bidireccionales
- Transmisión de datos de hasta 75 Kbps
- Los paquetes de datos contienen código de detección de errores. Además la capa física está optimizaba para un bajo consumo y para poder realizarlo a bajo coste

Control de Acceso al Medio (MAC)

- Permite a un dispositivo comunicarse con múltiples dispositivos, hasta 8 simultáneamente.
- Proporciona tiempo de acceso muy rápido y baja latencia.

Control de enlace lógico (LLC)

Proporciona características fiables para asegurar la secuencia de paquetes y las retransmisiones cuando se detectan errores.

En las siguientes secciones se van a describir algunos experimentos realizados sobre la tecnología IrDA utilizando diversos dispositivos.

2.6.3. Configuración de un dispositivo dongle

Normalmente, los ordenadores de sobremesa no cuentan con dispositivos de infrarrojos integrados como los ordenadores portátiles, por tanto, es necesario el uso de periféricos que provean de este tipo de conexión al ordenador. El dispositivo que hemos utilizado para realizar esta labor ha sido un dongle ACTISYS IR 220L plus, este dispositivo se conecta al puerto serie del ordenador.

Para configurarlo hay que realizar varias tareas:

Configuración del kernel

El kernel utilizado para la pruebas ha sido un kernel de la serie 2.4.x, aunque con la serie 2.2.x debería funcionar, pero en este último caso quizás haya que aplicar algún parche al kernel.

Es necesario activar ciertas opciones en el kernel, en el apartado IrDA (infrared) Support, y lo mejor es introducir estas opciones como módulos, y cargarlos solo cuando vayamos a usar el dispositivo infrarrojos. Las opciones básicas de IrDA que se deben activar son las siguiente:

- CONFIG_IRDA
- CONFIG_IRLAN
- CONFIG_IRCOMM
- CONFIG_IRDA_ULTRA
- CONFIG_IRDA_OPTIONS

Las opciones para usar un dispositivo de infrarrojos conectado al puerto serie:

- CONFIG_IRTTY_SIR
- CONFIG_IRPORT_SIR

Y la opción específica para el dispositivo que hemos utilizado nosotros es el siguiente:

- CONFIG_ACTISYS_DONGLE

Software

Después de esto, lo siguiente sería instalar el software necesario. Lo básico sería el paquete irda-common, que nos proporciona algunas herramientas como *irmanager* o *irattach*, que nos ayudarán después a configurar nuestro dispositivo.

También se debe instalar el paquete irda-tools, que nos proporciona herramientas como *irdadump* o *irdaping*, que nos ayudarán a depurar y a comprobar el funcionamiento del dispositivo.

Configuración de los módulos

Lo siguiente sería configurar los módulos para que se carguen solos cuando el kernel los necesite. Para ello hay que configurar los alias para el /etc/modules.conf.

Creamos un fichero, por ejemplo, /etc/modutils/irda y metemos lo siguiente:

```
#modutils/irda
alias tty-ldisc-11 irtty
alias char-major-161 ircomm-tty
alias char-major-60 ircomm_tty
alias char-major-10-187 irnet
#for dongle
alias irda-dongle-2 actisys
alias irda-dongle-3 actisys+
```

Después de esto hay que ejecutar *update-modules* para que se actualice el fichero /etc/modules.conf

Creación de los dispositivos

A continuación necesitamos crear los dispositivos que va a utilizar el dispositivo para comunicarse. Para ello ejecutamos los siguientes comandos:

```
mknod /dev/ircomm0 c 161 0
mknod /dev/ircomm1 c 161 1
mknod /dev/irlpt0 c 161 10
mknod /dev/irlpt1 c 161 11
mknod /dev/irnet c 10 187
```

Funcionamiento del dispositivo de infrarrojos

Una vez que hemos seguido todos estos pasos ya tenemos el PC configurado para poder usar el dispositivo de infrarrojos.

Para comprobar que hemos seguido todos los pasos bien, y que está bien configurado debemos ejecutar el siguiente comando.

irattach /dev/ttyS0 -d actisys -s

donde /dev/ttyS0 indica el puerto serie al que tenemos conectado el dispositivo, si lo tenemos conectado a otro puerto serie debemos cambiar este parámetro.

Después de esto podemos ejecutar lsmod y comprobar que los siguiente módulos se han cargado bien:

- actisys
- irtty
- irda

Para hacer esto último también se puede utilizar el script creado en /etc/init.d/irda, para poder cargar los módulos cuando los necesitamos y descargarlos cuando ya no nos hagan falta.

Y después para comprobar si estos dispositivos emiten o reciben algo podemos configurar dos PCs, poner los dos dispositivos uno enfrente del otro, y ejecutar *irdadump* en cada uno de los dos PCs. Con esto veremos los mensajes que emite y que recibe cada uno de los dispositivos ...

Conexión de dos ordenadores mediante TCP/IP sobre IrDA

La conexión de los dos ordenadores mediante TCP/IP sobre Irda es muy sencilla. Debemos tener cargados los módulos necesarios para usar los dispositivos como se indicaba en los apartados anteriores, y se deben poner un dispositivo enfrente del otro, cada uno conectado a uno de los dos ordenadores.

Primero debemos asegurarnos de que tenemos disponibles las siguientes opciones en el kernel, bien incluidas dentro de él o como módulos :

- IRNET
- PPP_GENERIC
- PPP_ASYNC
- PPP_DEFLATE

Y después de esto se ejecuta lo siguiente en el ordenador uno:
pppd /dev/irnet 9600 local noauth dirIPpc1:dirIPpc2
Y esto en el ordenador dos:
pppd /dev/irnet 9600 local noauth dirIPpc2:dirIPpc1
Se sustituirá dirIPpc1 y dirIPpc2 por las direcciones IP del ordenador uno y del ordenador dos respectivamente.

Cuando hemos ejecutado estos comandos se debe haber creado un dispositivo de red llamado ppp0 con la dirección IP que le hemos puesto en cada uno de los PCs. El parámetro 9600 indica la velocidad de la conexión, podemos cambiar este parámetro y ponerle mayor velocidad, hasta 115200 ...

Y una vez que ya tenemos esto, tenemos disponible una conexión TCP/IP entre los dos PCs mediante los dispositivos infrarrojos.

Podemos probar a hacer ping entre las dos máquinas, conexiones ssh, conexiones http, y todo lo que se nos ocurra ...

2.6.4. Configuración de IrDA en un HP Jornada 548

Para las pruebas hemos utilizados dos PDAs distintos, uno de ellos es un HP Jornada 548, que dispone de procesador a 133MHz 32-bit Hitachi, 32 Mb de memoria RAM, pantalla de 240x320 pixels en color, puerto IrDA, USB, Serie, ... Este PDA corre sistema operativo Windows CE.

Las pruebas con este dispositivo han sido las de conectarlo a un ordenador con una conexión TCP/IP sobre Irda. Para ello son necesarios los siguientes pasos:

Configuración del ordenador

En esta ocasión, el ordenador equipado y configurado como se explica en la sección 2.6.3, debe configurarse especialmente para hacer de servidor de la conexión ppp (Point-to-point) que se va a crear entre él y el PDA. Los pasos necesarios son los siguientes:

1. Crear un fichero /usr/sbin/cebox.sh que contenga lo siguiente:

   ```
   #!/bin/sh
   pppd call cebox
   ```

   Este fichero será el que tendremos que ejecutar para lanzar el servidor de la conexión en el ordenador.

2. Crear un fichero /etc/ppp/cebox.chat con lo siguiente:

   ```
   AT OK
   AT OK
   AT OK
   AT OK
   AT OK
   ATDT CONNECT
   ```

   En este fichero se indican las opciones de autentificación entre ambos (PC y Jornada).

3. Y por último hay que crear un enlace de /dev/irnine a /dev/ircomm0, para ello:

   ln /dev/ircomm0 -s /dev/irnine

4. Y crear el siguiente archivo /etc/ppp/peers/cebox con lo siguiente:

   ```
   /dev/irnine 115200 nocrtscts
   connect /usr/sbin/chat -v -f /etc/ppp/cebox.chat
   noauth
   local
   dirIPpc:dirIPjornada
   ms-dns servidorDNS
   ```

   Y en este último fichero se indican algunas opciones de la conexión. Habrá que sustituir dirIPpc y dirIPjornada por las direcciones IP del ordenador y de la HP Jornada respectivamente. Y también sustituir servidorDNS por la dirección IP del servidor de DNS disponible.

Configuración de la HP Jornada 548

Para la configuración del HP Jornada se deben seguir los siguiente pasos:

1. Primero se crea una nueva conexión por módem, para ello seleccionar la opción *Start/Settings* del menú, y dentro de esta opción seleccionar la pestaña que pone *Connections*. Se selecciona *Modem*.

2. Después se selecciona *New Connection ...*

3. Se selecciona como módem el *Generic Irda Modem*

4. Se elige la velocidad del módem a 115200, y se selecciona la opción *advanced ...*

5. Se deja todo como está excepto el *flow control* que se pone en *software*, y se pulsa OK.

6. Se selecciona *Next* ...

7. En el apartado de número de teléfono se pone todo a 0.

8. Se selecciona *Next* y después *Finish* ...

Con esto ya hemos configurado el Jornada para que pueda hacer de cliente de la conexión ppp sobre TCP.

Funcionamiento de la conexión

Cuando hemos realizado los pasos anteriores ya tenemos preparados el PC y el Jornada para la conexión TCP/IP.

Primero ejecutamos */usr/sbin/cebox.sh*, y ya estaría dispuesto el ordenador con GNU/Linux para hacer de servidor de la conexión ppp.

Después de esto ejecutamos el cliente de la conexión en el Jornada, para ello seleccionamos la opción Start/programs del menú y seleccionamos el icono de Connections. Ahí aparecerá la conexión que hemos creado anteriormente y solo deberemos pulsar sobre esta conexión para que se inicie la conexión por ppp.

Y una vez terminado todo esto tenemos conectados al PC y al Jornada mediante TCP.

Para hacer pruebas podemos poner por ejemplo un servidor web en el PC, y probar a navegar por sus páginas desde el Jornada con el Internet Explorer, también podemos probar otras conexiones como ftp, telnet, ...

2.6.5. Configuración del puerto de infrarrojos en una Ipaq

El otro tipo de dispositivos PDAs utilizado han sido varios Compaq Ipaq H3100, equipados con procesador StrongARM a 206 MHz, 64 Mb de memoria, pantalla a color de alta calidad (320x240), puerto IrDA, ... En estos dispositivos viene de serie el sistema operativo Windows CE, pero nosotros hemos instalado el sistema operativo Familiar [Fam01], que es un sistema operativo basado en Debian GNU/Linux pero para equipos sobre plataforma StrongARM.

La distribución de Familiar 0.4 que hemos usado tiene kernel 2.4.3, y junto con la distribución vienen unos paquetes para usar Irda.

Estos paquetes son: *irda-common* e *irda-modules-2.4.3.rmk2-np1*

Y para instalarlos hay que hacer:

- *ipkg install irda-common*
- *ipkg install irda-modules-2.4.3.rmk2-np1*

Después de esto habría que ejecutar *depmod -ae* porque el segundo de los paquetes añade al kernel los módulos necesarios para usar Irda.

Y solamente con esto ya tenemos preparado el Compaq Ipaq para que utilice el puerto de infrarrojos. Para activarlo haremos:

ifconfig irda0 up; echo 1 ¿/proc/sys/net/irda/discovery

Y para desactivarlo:

ifconfig irda0 down; echo 0 ¿/proc/sys/net/irda/discovery

2.7. Otras tecnologías

Además de estos tres tipos de tecnología descritos en los apartados anteriores (802.11, Bluetooth e Irda), existen muchos más protocolos para proporcionar comunicaciones inalámbricas, pero estos tres anteriores son los más usados y los que parecen tener un mayor futuro.

Los siguientes protocolos también facilitan la comunicación inalámbrica entre dispositivos:

- **HiperLAN**: Soporta canales de datos de 23.5 Mbps
- **HiperLAN2**: funciona en la banda de frecuencia de los 5 Ghz, que puede ser usada libremente en Estados Unidos y Asia, pero no en Europa. En Europa utiliza otra frecuencia de libre uso.
- **DECT**: Soporta canales básicos de 64 o 96 kbps.
- **RadioLAN**: Opera en la banda de frecuencia de los 5.8 Ghz.

2.8. Redes ad-hoc

Una red inalámbrica ad-hoc es una colección de dos o más dispositivos equipados con posibilidad de realizar comunicaciones inalámbricas para crear redes. Estos dispositivos se pueden comunicar con los demás, si están en su rango de cobertura o incluso si no está en él. En este último caso otro dispositivo debe hacer de intermediario.

Una red ad-hoc es **auto-organizada** y **adaptativa**. Esto significa que la red puede crearse o deshacerse sin necesidad de ningún sistema de administración. El término ad-hoc significa que puede tener diversas formas y puede ser móvil, individual o en red. Los nodos o dispositivos ad-hoc deben ser capaces de detectar la presencia de otros dispositivos y de realizar el protocolo de bienvenida a estos dispositivos para permitir la comunicación con ellos y poder compartir información y servicios.

Los dispositivos ad-hoc puede ser muy diversos (PDAs, ordenadores portátiles, teléfonos con conexión inalámbrica, ...), por tanto la capacidad de almacenamiento, de computación y las características técnicas de cada uno de ellos pueden varias muchísimo. Los dispositivos ad-hoc no solamente deben detectar la presencia de sus dispositivos colindantes, sino que debe identificar de qué tipo de dispositivos se trata y cuáles son sus atributos.

Como la red ad-hoc inalámbrica no se centra en ninguna entidad cableada, estos tipos de redes no poseen ninguna infraestructura creada anteriormente para poder realizar estas redes. Pero la información para encaminar paquetes a cada uno de los nodos debe cambiar y actualizarse para reflejar la posible movilidad de los dispositivos.

2.8.1. Heterogeneidad de dispositivos móviles

Como hemos dicho, los atributos de cada uno de los dispositivos de la red ad-hoc es importante, y estos dispositivos pueden tener variadas características, lo que afectará al rendimiento de las comunicaciones y al diseño de los protocolos de comunicación entre ellos.

La tabla 2.3 muestra las características de alguno de estos dispositivos. Hay diferencias en tamaño, capacidad de cómputo, memoria, disco, capacidad de las baterías, ...

La capacidad de un nodo ad-hoc para actuar como servidor o para ofrecer un servicio determinado dependerá de su capacidad de procesamiento, memoria y otros factores. Por esta razón habrá dispositivos en una red ad-hoc que podrán actuar como servidores o proveedores de servicios mientras que otros solo podrán actuar como clientes.

Uno de los factores más importantes de estos dispositivos es la capacidad de almacenar **energía**, porque normalmente las baterías de estos dispositivos son de poca duración, y las comunicaciones inalámbricas normalmente añade más consumo de energía. Por eso, cada dispositivo debe valorar su estado general antes de realizar tareas para otros nodos, como encaminador intermedio entre dos nodos o para ofrecer otros servicios.

Dispositivo	**Tamaño(cm)**	**CPU**	**Memoria(Mb)**	**Disco**	**Batería**
Palm Pilot	3.5x4.7	2.7	4-32	Ninguno	3-5.5
Teléfono móvil	2.5x5.5	16	1	Ninguno	3.6 V
Pocket PC	13x7.8	130-224	32-128	32 (ROM)	3.5
PC portátil	40x30	2000-2600	128-512	20-80	374-66

Cuadro 2.3: Características de algunos dispositivos existentes

2.8.2. Características especiales de las redes ad-hoc

Encaminamiento

La posibilidad de movilidad implica que los enlaces entre nodos pueden crearse y deshacerse a menudo y de forma no determinada. Por tanto, aunque se han desarrollado muchos algoritmos de encaminamiento para redes ad-hoc, muchos de estos protocolos no son eficientes cuando introducimos esta posibilidad de movilidad en los nodos, que puede ser muy frecuente y de forma no definida.

Es necesario desarrollar nuevos protocolos que sean adaptables a entornos en los que los nodos se mueven a menudo.

Multicast

El aumento del número de usuarios en Internet también se ha visto influido por la posibilidad de realizar conferencias de audio y de vídeo entre usuarios. Para la realización de estas conferencias entre varios usuarios son necesarios los protocolos de multicast. Este protocolo se basa en el *backbone*, que lo forman una serie de encaminadores multicast interconectados que son capaces de encaminar los paquetes multicast mediante túneles por los encaminadores que no son multicast.

La mayoría de protocolos multicast se basan en el hecho de que los encaminadores son estáticos, y que los nodos no se van a mover cuando se han establecido las comunicaciones. Por eso es necesario adaptar estos protocolos a las redes ad-hoc.

Disponibilidad de energía

La mayoría de los protocolos de red no consideran el consumo de energía como un factor de diseño, ya que asumen que los encaminadores y nodos son estáticos y por tanto están conectados a la red eléctrica.

Pero los nodos móviles utilizan baterías, y el campo de las baterías de energía no ha avanzado a la velocidad del campo de la computación y proporcionan un tiempo de vida muy limitado.

Por eso, es necesario tomar decisiones para el ahorro de energía en los dispositivos de redes ad-hoc, sobretodo cuando estos dispositivos no actúan como clientes de las acciones realizadas por los usuarios sino cuando deben actuar como intermediarios entre otros dispositivos.

Rendimiento de TCP

TCP es un protocolo punto a punto, diseñado para el control de flujo y congestión en la red. TCP es un protocolo orientado a conexión, lo que implica

que hay una fase de establecimiento de conexión antes de transmitir los datos. TCP asume que los nodos entre el origen y destino son estáticos, y solo realiza control de flujo y congestión en el nodo origen y destino.

Desafortunadamente, TCP es incapaz de distinguir la presencia de movilidad y de congestión en la red. Por tanto la movilidad en los nodos puede provocar pérdida de paquetes y largos periodos de RTT (*round-trip time*). Por eso es necesario realizar algunos cambios para asegurar que el protocolo de transporte realiza correctamente su labor sin afectar el rendimiento de las conexiones punto a punto.

Servicios de localización y acceso

Al igual que los protocolos son importantes para el rendimiento de la red ad-hoc, los servicios que se pueden prestar en la red también son importantes, como servicio de localización y acceso.

Para la realización de estos servicios es necesario que se realice todavía mucha investigación, porque la infraestructura que ahora se utiliza en otras redes puede no ser la más adecuada. Probablemente la arquitectura tradicional cliente/servidor no es válida en estas redes, porque los clientes inician una petición de un servicio a otro nodo, y todos los nodos debido a sus características no están disponibles para realizar estas acciones.

2.8.3. Protocolos para redes ad-hoc

Como hemos dicho, las redes ad-hoc están formadas por una colección de nodos móviles que cooperan entre ellos, sin la necesidad de intervención de ningún punto de acceso centralizado. A continuación voy a mostrar el modo de funcionamiento de estas redes ad-hoc.

La idea básica de diseño de estas redes es que cada nodo actúa como un encaminador, y periódicamente anuncia la visión que él tiene sobre la topología de conexión de la red a los demás nodos móviles. Esta idea nos lleva a un nuevo tipo de protocolo de encaminamiento para estas redes ad-hoc.

Desde un punto de vista teórico, una red ad-hoc es un grafo [Per01], que está formado por los nodos (dispositivos móviles) y arcos entre estos nodos (cobertura para la comunicación entre dos nodos). La topología que puede tener este grafo puede ser arbitraria, porque no existen limitaciones sobre la situación de cada nodo con respecto a los demás.

Los protocolos diseñados para estas redes pueden ser clasificados en varias categorías [RT99] [Toh02] que presentamos a continuación junto con algunos ejemplos de protocolos de cada categoría:

Dirigido por tablas de encaminamiento

Los protocolos de encaminamiento de redes ac-hoc dirigidos por tablas intentan mantener consistente y actualizada la información de encaminamiento de las rutas desde cada nodo a los otros nodo de la red. Estos protocolos requieren que cada nodo mantenga una o más tablas con información de encaminamiento, y que respondan a los cambios de la topología de red propagando las rutas a través de la red para mantener la consistencia.

La diferencia entre ellos consiste en el número de las tablas de encaminamiento necesarias y el modo de actualizar la información con los cambios producidos en la red.

- **DSDV** (*Destination Sequenced Distance Vector*): Está basado en el algoritmo clásico de encaminamiento distribuido *Bellman-Ford* [Bel58]. La mejora realizada es evitar la creación de bucles en la red de encaminadores móviles (nodos). Cada nodo de la red mantiene una tabla de rutas con los posibles nodos destinos y el número de saltos hasta ellos.

  Las actualizaciones de las tablas son enviadas a la red para mantener la consistencia, pero para que estas operaciones no consuman más ancho de banda del necesario existen dos tipos de paquetes de actualización de rutas. Uno de ellos envía toda la información disponible sobre encaminamiento, y suele enviarse esporádicamente cuando hay escasos movimientos de red. El otro tipo de paquetes con actualización de rutas contiene solo la información que ha cambiado desde el envío del último paquete de este tipo.

- **WRP** (*Wireless Routing Protocol*):

  WRP mantiene cuatro tablas:

  - Tabla de distancia: indica el número de saltos entre un nodo y su destino
  - Tabla de encaminamiento: indica el siguiente salto hacia el destino
  - Tabla de coste de enlace: retardo asociado a una determinada ruta
  - Tabla de lista de retransmisión de mensajes: contiene un número de secuencia del mensaje de actualización, un contador de retransmisiones, un vector de confirmaciones y una lista de actualizaciones enviadas en un mensaje de actualización.

  Los nodos envían mensajes de actualización periódicamente. El mensaje contiene una lista de actualizaciones, y una lista de respuestas indicando qué nodos deben confirmar la actualización. Un nodo envía un mensaje de actualización después de procesar los mensajes de actualización de sus vecinos o cuando detecta cambios en algún enlace.

  Una parte original de este protocolo es la forma que tiene de encargarse de los bucles. En WRP los nodos anuncian la distancia y la información sobre el segundo salto para cada destino en la red inalámbrica. WRP pertenece a la clase de algoritmos de búsqueda de caminos pero con una importante excepción, evita el problema de *cuenta hasta el infinito* forzando a cada nodo a realizar comprobaciones de la información que le envía el nodo predecesor acerca de los vecinos.

- **CSGR** (*Cluster Switch Gateway Routing*): Los nodos son agrupados y cada grupo contiene un director. Con la creación de estos grupos se establece una forma de jerarquía, en los que cada director puede controlar su grupo de nodos ad-hoc.

  El director es elegido mediante un algoritmo distribuido, y cuando un director se mueve fuera del grupo, se elige otro director. El problema

puede surgir cuando el director cambia frecuentemente porque los nodos gastan mucho tiempo y recursos eligiendo al nuevo director.

CSGR usa DSDV como algoritmo base de encaminamiento, pero modificado para utilizar la jerarquía introducida de los directores y los grupos.

Por demanda iniciada en origen

Este tipo de encaminamiento crea rutas solamente cuando es solicitado por un nodo origen. Cuando un nodo necesita una ruta hacia un destino, inicia un proceso de descubrimiento de rutas dentro de la red. Este proceso termina cuando se han examinado todas las posibles rutas y se encuentra una ruta disponible. Cuando se ha establecido una ruta entre dos nodos, esta información es mantenida hasta que el destino pasa a ser inalcanzable desde el origen o cuando la ruta ya no es necesaria.

- **AODV**(*Ad Hoc On Demand Distance Vector Routing*): Este protocolo es una mejora del protocolo DSDV, porque AODV minimiza el número de mensajes broadcast por la red mediante el método de creación de rutas cuando son solicitadas por un nodo origen, en lugar de mantener una lista completa de rutas como en DSDV.

  Cuando un nodo quiere mandar un mensaje a un destino y no posee una ruta válida inicia un procedimiento de descubrimiento de ruta: envía un broadcast solicitando ruta (RREQ: *Route request*) a sus vecinos, estos reenvían el mensaje a sus vecinos, y así sucesivamente hasta que la solicitud llega al destino o a un nodo con información fiable de la ruta solicitada. Cuando el nodo destino o el nodo con información fiable recibe la solicitud envía un paquete de respuesta (RREP: *Route reply*) al nodo origen. Este paquete es enviado al nodo origen y es encaminado por el mismo camino que llegó el RREQ, porque cada nodo intermedio al que le llegó este primer mensaje almacenó el nodo por el que le había llegado, por tanto se puede reconstruir el camino inverso del RREQ para hacer llegar el RREP al nodo emisor. Cuando el nodo origen recibe el RREP ya puede mandar los datos al destino porque los nodos intermedios han aprendido la ruta.

  Cada paquete RREQ contiene un número de secuencia para evitar bucles si los nodos intermedios lo reciben más de una vez, y de este modo no los vuelven a reenviar. Los enlaces son simétricos, porque el RREP vuelve por donde llegó el RREQ, y los nodos intermedios guardan información sobre el siguiente salto para un destino en su tabla de rutas.

- **DSR** (*Dynamic source routing*): Este protocolo también está basado en el concepto de encaminamiento en origen. Los nodos deben mantener una caché de rutas que contienen las rutas en origen que conciernen a ese nodo.

  El protocolo consiste en dos fases:

  - Descubrimiento de rutas: Cuando un nodo quiere mandar un mensaje a otro y no posee una ruta válida, inicia un procedimiento de descubrimiento de rutas enviando mediante broadcast un paquete RREQ, este paquete contiene el nodo origen, el nodo destino y un identificador. Cuando los otros nodos reciben el paquete comprueban

si conocen la ruta solicitada, si no la conocen añaden su dirección al paquete y lo reenvían a sus vecinos.

Si el nodo que recibe el paquete conoce la ruta, coloca esta ruta en el paquete RREP y se lo envía al origen. Si el nodo que recibe el paquete es el destino, añade su dirección a la lista de nodos del RREQ, la copia en el paquete RREP y se la envía al origen. Para enviar el RREP hay dos opciones: si están disponibles los caminos simétricos se utiliza el camino inverso por el que ha llegado el RREQ, y sino se inicia un nuevo RREQ para hallar la ruta hacia el origen del primer RREQ.

- Mantenimiento de rutas: El mantenimiento de rutas se realiza mediante paquetes de errores en rutas (RERR) y confirmaciones. Cuando un nodo detecta error en un enlace envía un RERR, y los nodos que lo reciben borran las rutas que almacenan en sus cachés hacia ese nodo, o las truncan en la parte que aparece el enlace que no funciona. Los paquetes de confirmación son usados para verificar el estado de los enlaces entre nodos.

- **TORA** (*Temporally Ordered Routing Algorithm*): Este protocolo es muy adaptable, libre de bucles basado en los enlaces inversos. Está diseñado para funcionar en entornos muy dinámicos con alta movilidad en los nodos de la red. Es iniciado en origen y facilita múltiples rutas para un determinado origen-destino.

  La clave de este protocolo es la localización de mensajes de control en un conjunto muy reducido de nodos cercanos al cambio en la topología. Los nodos mantienen información de encaminamiento sobre los nodos vecinos más próximos, un solo salto.

  La clave de este protocolo es la reacción ante fallos de enlaces. La reacción ante este hecho es organizada mediante la difusión de secuencias de enlaces inversos. Cada enlace inverso consiste en la búsqueda de rutas alternativas para el destino. Esta búsqueda a menudo solo necesita la ejecución del procedimiento una sola vez, con lo que se reduce el número de mensajes necesarios para actualizar las rutas.

Híbridos

- **ZRP**(*Zone Routing Protocol*): Éste es un protocolo híbrido, que incorpora las ventajas de los protocolos por demanda y de los protocolos pro-activos.

  Cada zona de encaminamiento es similar a un grupo con la excepción de que cada nodo actúa como director y como miembro de otros grupos, las zonas se pueden superponer. El tamaño de las zonas afecta al rendimiento de las comunicaciones en la red.

  Cada zona contiene unos pocos nodos móviles con uno, dos o más saltos hasta el nodo central de la zona, dentro de cada zona el protocolo de encaminamiento utilizado es un protocolo dirigido por tablas de encaminamiento. Por tanto las actualizaciones de rutas se realizan dentro de la zona, y cada nodo tiene una ruta hacia todos los nodos dentro de la zona.

Cuando un nodo quiere comunicarse con otro que no está dentro de su zona, debe iniciar una búsqueda mediante un protocolo de encaminamiento por demanda.

2.8.4. Prácticas con redes ad-hoc

Como hemos dicho en los apartados anteriores, las redes ad-hoc están formadas por dispositivos móviles. Estos dispositivos móviles pueden ser ordenadores portátiles, PDAs, ... Pero también existen otro tipos de dispositivos móviles que pueden coexistir con los anteriores y que están provistos de funcionalidades totalmente distintas pero muy interesantes, y son los que se han utilizado para realizar prácticas sobre redes ad-hoc.

Estos dispositivos son pequeños robots, comercialmente conocidos como *Legos Mindstorms*, que disponen de un pequeño procesador (Hitachi 8300), motores y varios sensores. El sistema operativo de estos dispositivos es bastante limitado a la hora de programar sobre ellos aplicaciones con cierto grado de complejidad, por ello, existe otro sistema operativo alternativo llamado LegOS, que se puede instalar en estos robots y tiene las siguientes características:

- Capacidades para ejecutar varias tareas
- Mecanismos de ahorro de energía
- Gestión dinámica de memoria
- Uso de semáforos
- Acceso a los cada uno de los elementos de los robots:
  - *Display*
  - Botones del *RCX*
  - Motores
  - Sensores
  - ***Comunicación por Infrarrojos***

Precisamente el último punto, la *comunicación por infrarrojos*, va a permitir a los robots ejecutar un protocolo de *encaminamiento ad-hoc* simplificado de forma que desde cualquier robot se puedan establecer comunicaciones con cualquier otro robot dentro de la *red ad-hoc*.

Diseño de la práctica con redes ad-hoc

Lo que se ha pretendido con la práctica es lo siguiente: Tenemos un conjunto de robots con capacidad de comunicaciones por infrarrojos, y vamos a implementar un protocolo para redes ad-hoc de los que hemos descrito en la sección 2.8.3, para que este grupo de robots sea capaz de comunicarse entre ellos, puedan crear una red ad-hoc solamente descubriendo cada uno de ellos a sus vecinos, y permitan la movilidad de los mismos, con lo que se crearán nuevas rutas y se eliminarán otras existentes.

Ya que la interacción con estos robots es compleja, porque el sistema de entrada/salida es muy limitado, se va a crear también una capa de software

sobre el protocolo implementado para su posible uso desde un ordenador, y poder interactuar con estos robots desde él. Se va a crear un interfaz web en el ordenador, para que desde él se pueda transmitir información a los nodos que estén en la red ad-hoc.

Por tanto, el objetivo final va a ser, que a cada nodo se le va a asignar un identificador, que va a ser la dirección que les va a identificar unívocamente en la red. Desde el interfaz web del ordenador se le van a enviar datos a uno de los robos que esté en la red y mediante el protocolo de comunicaciones implementado los datos le llegarán al dispositivo y lo mostrará por la pequeña pantalla de que disponen.

El protocolo elegido para la implementación ha sido **DSR**.

Implementación del protocolo DSR

Las funciones implementadas del protocolos son las siguientes:

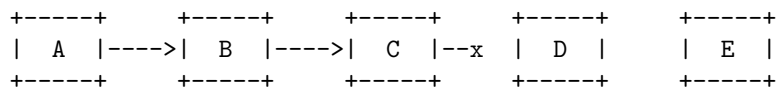
- **Route Discovery**:

  Mediante esta función, el nodo que quiere enviar un mensaje inicia el proceso para calcular la ruta hacia el destino. Mediante el envío de paquetes RREQ por la red, y mediante el reenvío de este paquete por los nodos intermedios al final recibirá un mensaje RREP con la información de la ruta que necesitaba.

  Las rutas utilizadas son simétricas, por tanto el mensaje RREP vuelve por el mismo camino por el que llegó el RREQ.

- **Route Maintenance**:

  Al originarse o encaminar un paquete utilizando una ruta en origen, cada nodo que transmite el paquete es responsable de la confirmación de que el paquete ha sido recibido por el siguiente salto de la ruta en origen. En el ejemplo siguiente, el nodo *A* desea enviar un paquete al nodo *E* utilizando ruta en origen a través de los nodos intermedios *B*, *C* y *D*.



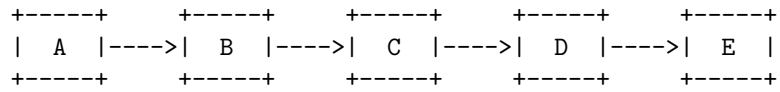
  En este caso, el nodo *A* es responsable de la recepción del paquete en el nodo *B*, el nodo *B* es responsable de la entrega del paquete en el nodo *C*, el nodo *C* de la recepción en el nodo *D*, y el nodo *D* es responsable de la recepción final en el nodo destino *E*.

  Si no se recibe confirmación de que el paquete ha sido recibido en el nodo siguiente de la ruta, tras un número máximo de intentos, el nodo que es responsable de la entrega del paquete en el siguiente nodo debería devolver un *Route Error* al nodo emisor del paquete indicando el enlace por el cual no ha podido ser encaminado el paquete. En el ejemplo anterior, el nodo *C* no puede entregar el paquete en el siguiente nodo, el *D*, y devuelve un *Route Error* a *A* indicando que el enlace entre *C* y *D* está caído. El nodo *A* eliminará de su *Route Cache* el enlace para ese nodo.

- Actualización de *Route Caches* en nodos intermedios

  Un nodo que está en la ruta en origen de un paquete, puede añadir información de encaminamiento al encaminar los paquetes hacia el nodo destino a su propia *Route Cache* de cara a poder utilizarla en futuros envíos.

  Por ejemplo, el nodo A utiliza una ruta en origen para comunicarse con E.

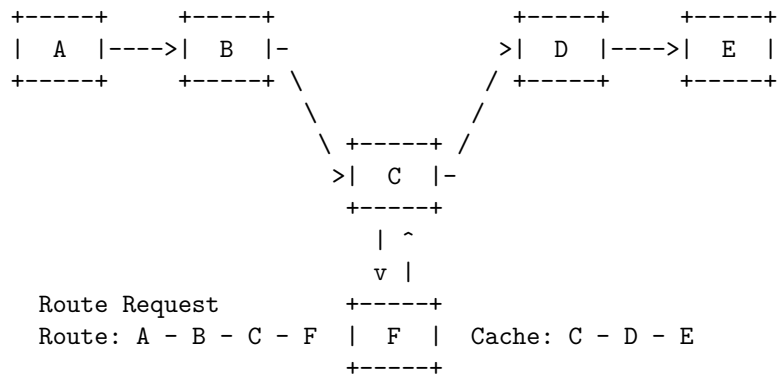


  Como el nodo C encamina el paquete en la ruta desde A hasta E, puede actualizar su cache con la información de las rutas hacia A y hacia E.

- Respuestas a las *Route Requests*

  Cuando un nodo recibe un *Route Request* para la cual él no es el destino de la misma, busca en su *Route Cache* a ver si existe una ruta para el nodo destino de la petición. Si la encuentra, el nodo generalmente devuelve un *Route Reply* al emisor en vez de continuar enviando la solicitud hasta el nodo destino. En la *Route Reply*, este nodo actualiza la lista de nodos que tiene que seguir el paquete en su camino hasta el destino, concatenando los que ya traía con los de la ruta que tenía en su cache.

  Sin embargo, antes de transmitir la *Route Reply*, el nodo debe verificar que la lista resultante a enviar en la *Route Reply* no contenga nodos duplicados. Por ejemplo, la siguiente figura ilustra una situación en la que la *Route Request* para el destino E ha sido recibida por el nodo F, y el nodo F ya tiene en su *Route Cache* una ruta de él mismo hasta el nodo E:



  La concatenación de la lista de nodos que traía la *Route Reply* con la ruta que el nodo F tenía en su *Route Cache* produce un nodo duplicado al pasar de C a F y de F a C nuevamente.

Implementación del interfaz web

El ordenador va a ser el encargado de interactuar a través de infrarrojos con la *red ad hoc* que formen los *Lego Mindstorms*. El PC va formar parte de la red, siendo un nodo *"especial"* encargado de iniciar las peticiones al resto de nodos.

Las interacción del PC con el usuario se realizarán a través de *HTTP*, por lo que el programa que se ejecute en el PC será un servidor que responda a peticiones *HTTP*.

Además el ordenador tiene que utilizar también el protocolo utilizado para comunicarse con los robots de la red ad-hoc, ya que él también forma parte de esta red. Y este nivel de protocolo ad-hoc en el ordenador irá acompañado de la librería que proporcionan los legos para la comunicación por infrarrojos desde el ordenador.

En el PC por tanto vamos a tener los siguientes elementos:

- Servidor HTTP.
- Nivel DSR.
- Demonio LNP.

Por tanto el funcionamiento será: El ordenador recibe una petición a través del servidor web, la procesa y efectúa las llamadas pertinentes al protocolo DSR implementado para comunicarse con los robots.

Maqueta de pruebas

Dados una serie de *Legos Mindstorm* (5 o 6 aproximadamente) separados por distancias considerablemente significativas, se han de poder comunicar todos con todos usando la versión simplificada del protocolo *DSR*.

La figura 2.2 ilustra un posible escenario en el que se desarrollarían las pruebas y el flujo de control que seguiría cada elemento:

Inicialmente, cada robot está identificado con un número que se utilizará en los paquetes, en los campos de dirección origen y de dirección destino (a modo de dirección *IP*).

El ordenador conoce a priori cuántos nodos pueden formar la red Ad Hoc, de forma que en cualquier momento, desde el interfaz web se puede enviar información a cualquiera de ellos.

Cada robot posee un radio de acción limitado, por lo que mediante los mecanismos de encaminamiento Ad Hoc de la versión simplificada del protocolo DSR, se puede alcanzar cualquier nodo (siempre que éstos estén disponibles).

Al comenzar, todos los robots tienen sus *Route Caches* vacías. Desde la interfaz web se solicita, por ejemplo, mandar información al robot número *5*. Esta petición le llega al ordenador que se la hace llegar al robot número *1* que es el que hace de enlace entre la red Ad Hoc y el ordenador.

El primer paso para el robot *1* es consultar su *Route Cache*. Como no tiene ninguna entrada para el nodo *5* solicitado, inicia el mecanismo de *Route Discovery* y envía un mensaje de *broadcast* con el *Route Request Packet*. En la parte de datos indicará, aparte del identificador de solicitud y el nodo destino, la lista de nodos por los que ha pasado el mensaje y que inicialmente solo contendrá un elemento, el *1* correspondiente al nodo solicitante de la ruta.

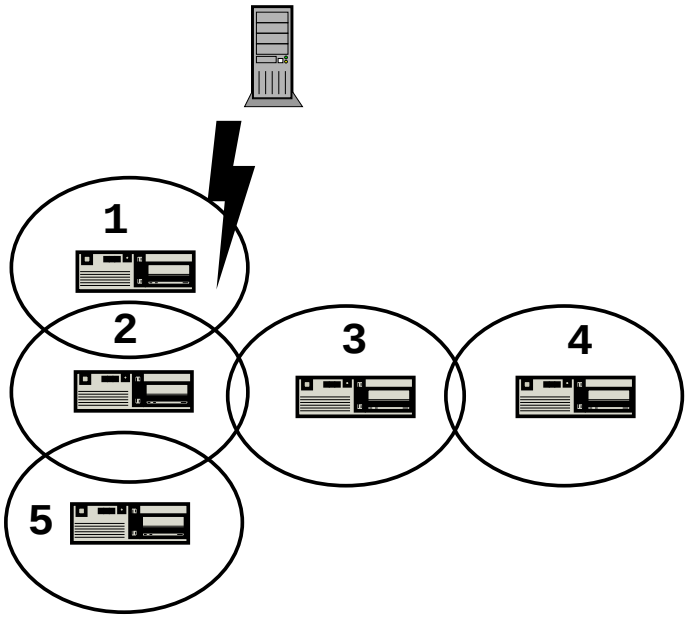


Figura 2.2: Maqueta red ad-hoc. Situación inicial

Este paquete es recibido por el nodo *2*, que al no ser el nodo destino solicitado retransmitirá el paquete. De esta forma, con sucesivas retransmisiones, el paquete se difundirá por la red, y cada nodo que no sea el nodo destino de la solicitud, se añadirá a la lista de nodos y retransmitirá el paquete.

De esta forma, la retransmisión de la solicitud del robot *2*, alcanzará al robot *5*. Éste, al ser el nodo destino de la solicitud enviará un *Route Reply* al nodo *1* utilizando la ruta recién creada para hacerle llegar el paquete. Por tanto, en la parte de datos se indicará el identificador de la solicitud a la que se contesta y la ruta que ha de seguir.

La nueva ruta creada (`1-2-5`) será utilizada por los nodos que la componen y a través de los cuales pasará el *Route Reply Packet*, para actualizar sus *Route Caches*. De esta forma, se tienen las siguientes entradas en cada uno de estos nodos:

- *Robot #1*:

Nodo destino	*Ruta*
2	1
5	2

- *Robot #2*:

Nodo destino	*Ruta*
1	2
5	2

- *Robot #5*:

Nodo destino	*Ruta*
1	2
2	5

En la figura 2.3, el robot *5* ha cambiado de situación y ya no está bajo el radio de acción del robot *2*.

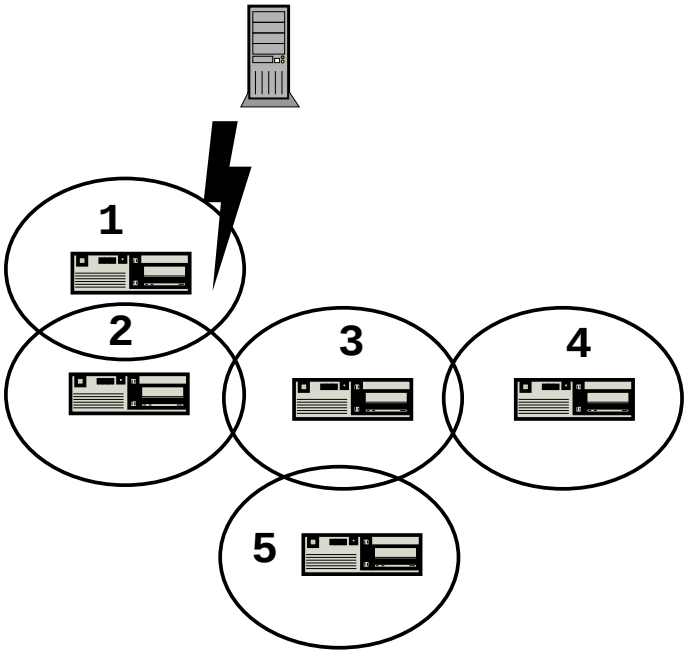


Figura 2.3: Maqueta red ad-hoc. Situación 2

Nuevamente se desea mandar información al robot *5*. El nodo *1* consulta su *Route Cache* y ve que para enviar un paquete al nodo*5* tiene que pasar por el nodo *2*, por lo que envía un paquete dirigido al nodo siguiente en la lista de nodos a atravesar (*2*) para hacerle llegar la información junto con dicha lista (*1-2-5*).

Al llegar el mensaje al robot *2*, este lo envía al siguiente nodo que es a la vez el nodo destino, el robot *5*. Sin embargo, dicho robot ya no recibe el mensaje porque ya no está en el radio de acción del nodo *2*, por lo que tras no una serie de reintentos del nodo *2* de hacerle llegar el mensaje y ver que no obtiene el *Route ACK Packet* correspondiente (que informa de que el paquete ha sido entregado correctamente), envía un *Route Error Packet* al nodo origen, el *1*, para indicarle que el nodo *5* ya no es alcanzable desde el nodo *2*.

Al recibir el *Route Error* el nodo *1*, iniciará nuevamente el proceso de *Route Discovery*, siguiendo los mismos pasos que para la figura 2.2.

En este caso, el robot *3* es el nuevo (y único) enlace hacia el nodo destino *5*. Las *Route Caches* de los nodos son nuevamente actualizadas quedando como sigue:

- *Robot #1*:

Nodo destino	*Ruta*
2	1
3	2
5	2-3

- *Robot #2*:

Nodo destino	*Ruta*
1	2
3	2
5	3

- *Robot #3*:

Nodo destino	*Ruta*
1	2
2	3
5	3

- *Robot #5*:

Nodo destino	*Ruta*
1	3-2
2	3
3	5

Otra posible situación representaría la que se muestra en la figura 2.4: Las *Route Caches* quedarían de la siguiente manera:

- *Robot #1*:

Nodo destino	*Ruta*
2	1
3	2
4	2-3
5	2-3-4

- *Robot #2*:

Nodo destino	*Ruta*
1	2
3	2
4	3
5	3-4

- *Robot #3*:

Nodo destino	*Ruta*
1	2
2	3
4	3
5	4

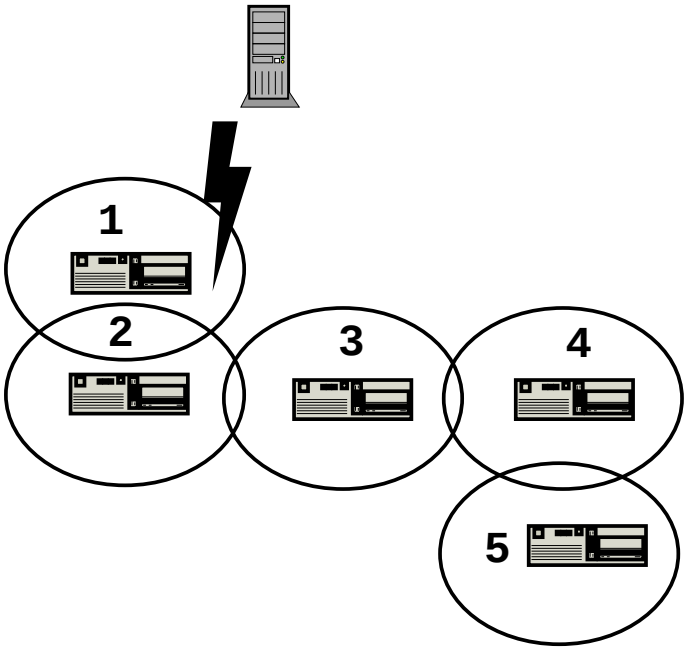


Figura 2.4: Maqueta red ad-hoc. Situación 3

- *Robot #4*:

Nodo destino	*Ruta*
1	3-2
2	3
3	4
5	4

- *Robot #5*:

Nodo destino	*Ruta*
1	4-3-2
2	4-3
3	4
4	5

Capítulo 3

Movilidad

3.1. Introducción a la movilidad sobre TCP/IP

Como hemos explicado en el capítulo anterior, el aumento de uso de los dispositivos inalámbricos cada día es más frecuente y nos ofrece otra perspectiva de uso de los dispositivos electrónicos, ordenadores personales, agendas electrónicas y otros dispositivos móviles.

En esta nueva perspectiva uno de los factores más relevantes es que el uso de estos dispositivos ya no se debe realizar desde un lugar estático, como anteriormente, sino que ahora nos ofrecen la posibilidad de moverse con nosotros, nos ofrecen una posibilidad de movilidad de dispositivos junto con los usuarios.

Para que esta movilidad ofrezca un servicio realmente útil para los usuarios no debe ser solamente que los dispositivos puedan moverse de un lugar a otro, sino que la posibilidad de movimiento se realiza junto con las comunicaciones que esté llevando a cabo el dispositivo. Necesitamos que los dispositivos tengan la posibilidad de movilidad entre redes.

Las redes sobre las que se mueven estos dispositivos son redes que funcionan sobre la pila de protocolos TCP/IP, y por eso el proporcionar **movilidad** a los dispositivos sobre estas redes no es algo inmediato. La base de las comunicaciones sobre TCP/IP es que cada nodo dispone de una dirección con la que se le identifica unívocamente, y esta dirección sirve para encaminar todas las comunicaciones que se generan desde y hacia él.

Esta dirección que cada nodo posee es asignada dependiendo de la red a la que esté conectado, y depende directamente de ella. Si un nodo cambia de red a la que está conectado y continúa con la dirección IP de la otra red no podrá comunicarse más, porque las paquetes de datos son encaminados hacia la red a la que pertenece esa dirección.

Por todo ello, necesitamos algún mecanismo que proporcione la posibilidad de moverse entre redes para estos dispositivos inalámbricos y que puedan seguir comunicándose normalmente y de forma transparente para el usuario y las aplicaciones independientemente de la red a la que esté conectado en cada momento.

Estos mecanismos son los mecanismos de movilidad, que se pueden englobar en dos tipos:

- *Micro-movilidad*: la movilidad que se realiza entre varios puntos de acceso

- *Macro-movilidad*: la movilidad que se realiza entre varias subredes de una misma organización

Estos dos tipos de protocolos de movilidad intentan solucionar el mismo problema, pero desde diferentes grados de la movilidad necesaria por los nodos móviles. Existen diferentes protocolos para cada uno de estos grupos de movilidad:

- Protocolos de micro-movilidad:
  - Mobile IP
- Protocolos de macro-movilidad:
  - Cellular IP
  - Hawaii
  - Hierarchical Mobile IP

De estos protocolos existen varias implementaciones, pero los protocolos más usados y que parece que será los que pasarán a ser usados son: **Mobile IP** y **Cellular IP**. Estos protocolos será los que describiré en las próximas secciones, y los que hemos usado para realizar los ensayos prácticos.

En la tabla 3.1 se muestran algunas de las implementaciones de estos protocolos y sus características.

Protocolo	Proyecto	Sistema operativo	Última versión	Licencia	Lenguaje de prog
MobileIPv4					
	Mobile IP at NUS	Linux 2.0.34	3.0beta	GPL	C
	Monarch	NetBSD 1.1 FreeBSD 2.2.2	1.1.0	FreeBSD-like	C
	Secure Mobile	FreeBSD 4.5 Linux 2.2.12-20	4.5	FreeBSD-like	C
	Solaris	Solaris 2.5.1, 2.6	?	Solaris License	C
	Linux Mob IP	Linux 2.2.x	2.0.2beta	FreeBSD-like	C
MobileIPv6					
	Mobile IP at NUS	Linux 2.1.59	1.0alpha	GPL	C
	Monarch	FreeBSD 2.2.2	1.0	BSD-like	C
	MIPL	Linux 2.4.x	0.9.1	GPL	C
	Lancaster	Linux 2.1.9x	0.4beta	Non *libre*	?
CellularIP					
	Columbia	Linux 2.2.14 FreeBSD 3.2	1.1(Linux) 1.0(FreeBSD)	Prop	C
Hierarquical Mobile IP					
	Dynamics	Linux 2.2.x, 2.4.x	0.8.1	GPL	C
Hierarchical Mobile IPv6					
	Inria	FreeBSD 3.4	2.0	BSD-like	C

Cuadro 3.1: Implementaciones de protocolos de micro/macro movilidad

3.2. Mobile IP

3.2.1. Fundamentos de Mobile IP

Mobile IP [Per96] es una modificación del protocolo IP que permite a los nodos continuar recibiendo datagramas independientemente de la red a la que estén conectados, esto conlleva mensajes de control adicionales que permiten manejar el encaminamiento de los datagramas. Este protocolo ha sido diseñado con la premisa de que debe ser escalable, porque se espera que en el futuro un gran porcentaje de los nodos conectados a Internet sea móvil.

El protocolo IP asume que una dirección IP identifica unívocamente el punto de conexión a la red de un nodo. Antes de que un nodo pueda recibir datagramas, ese nodo debe ser identificado en la red en la que está conectado, sino el nodo será inalcanzable. Si no utilizamos Mobile IP, uno de los dos mecanismos siguientes podría ser utilizado para que un nodo cambie de punto de conexión a la red sin perder la conexión a la red:

- Un nodo puede cambiar su dirección IP cada vez que cambia de punto de conexión a la red
- Rutas específicas para cada nodo pueden ser propagadas por la parte de infraestructura de red relevante

En general, cualquiera de estas soluciones son inaceptables. Con la primera es imposible que el nodo mantenga la conexión del nivel de transporte y superiores cuando el nodo cambia el punto de conexión. Y la segunda solución tiene claros problemas para poder escalar, más aún cuando aumente el número de dispositivos móviles.

Los siguientes objetivos son los básicos que cualquier implementación de Mobile IP [Per97] debe cumplir:

- Un nodo móvil debe ser capaz de comunicarse con otros nodos después de cambiar su punto de conexión a la red, incluso sin cambiar su dirección IP.
- Un nodo móvil debe ser capaz de comunicarse con otros nodos que no corren ninguna implementación de Mobile IP. No es necesario que los nodos o encaminadores que no cumplen alguno de las funciones del mecanismo de Mobile IP tengan ninguna característica específica en su pila de protocolos.
- Los mensajes que se envían para informar del punto de conexión a la red de un nodo deben ser autentificados, para protegerse contra ataques por redirecciones.
- Normalmente el medio de enlace de los dispositivos móviles es inalámbrico, lo que conlleva un menor ancho de banda y una mayor tasa de errores que un enlace por cable. Por tanto, el número de mensajes de control que son enviados al nodo móvil debe ser minimizado al máximo, y el tamaño de estos mensajes debe ser tan pequeño como sea posible.
- Mobile IP no debe poner restricciones a la hora de asignar direcciones IP a los nodos móviles, cada organización debe asignar las direcciones que le pertenezcan.

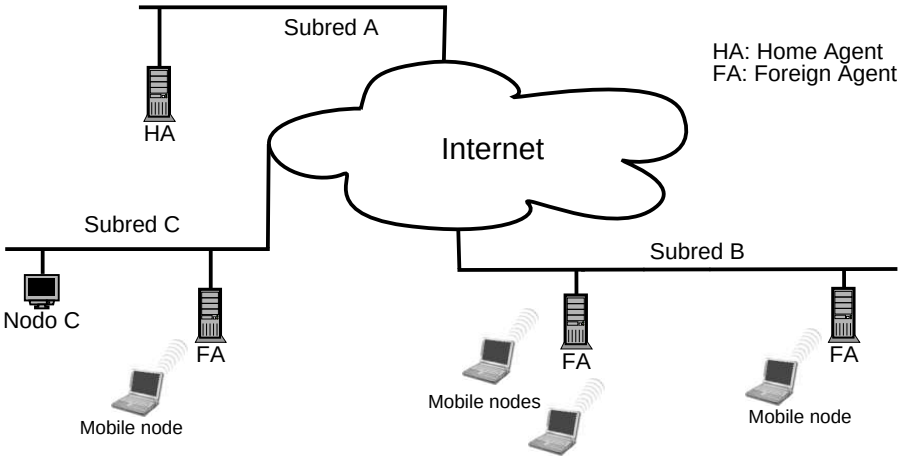


Figura 3.1: Infraestructura de nodos en Mobile IP

Mobile IP habilita a los nodos móviles para moverse de una subred a otra, y esta operación puede realizarse entre medios heterogéneos u homogéneos. Es decir, un nodo puede moverse de una red Ethernet a otra, de una red Ethernet a otra inalámbrica, ...

3.2.2. Infraestructura de Mobile IP

Mobile IP necesita que una serie de nodos tengan una funcionalidad especial para que tener la infraestructura necesaria:

- ***Mobile Node***: Es un nodo móvil que cambia su punto de conexión de una red o subred a otra. Puede cambiar su posición sin cambiar su dirección. Puede comunicarse con otros nodos desde cualquier punto de conexión usando su dirección IP.
- ***Home Agent***: Es un encaminador de la red original (*home network*) a la que pertenece el *mobile node*, que se encarga de encaminar los paquetes hacia él cuando éste no se encuentra en su *home network*, porque mantiene la información de la posición actual del *mobile node*
- ***Foreign Agent***: Es un encaminador de la red a la que se mueve el *mobile node*, llamada *visited network*, que le proporciona servicios de encaminamiento mientras éste se está registrando. El *foreign agent* recoge los datagramas que envía por un túnel el *home agent* y se los entrega al *mobile node*

En la figura 3.1 se muestra un ejemplo de una infraestructura creada para el uso de Mobile IP.

El *mobile node* siempre posee una dirección de su red original (*home address*). Cuando está fuera de su *home network*, una dirección de la red que visita le es asignada (*care-of address*) y refleja en cada momento el punto de conexión actual del *mobile node*. El *mobile node* usa su *home address* como la dirección origen de todos los datagramas que envía, excepto durante el proceso de registro si tiene que adquirir una nueva dirección.

El protocolo Mobile IP se basa fundamentalmente en la realización de las siguientes funciones:

- ***Agent discovery***: *home agents* y *foreign agents* pueden advertir su disponibilidad por cada uno de los enlaces por los que prestan servicio. Y también un *mobile node* puede mandar solicitar información sobre los agentes que existen.
- **Registro**: Cuando el *mobile node* se encuentra fuera de su *home network*, éste registra su dirección adquirida al *home agent*. Dependiendo del método de conexión, se registrará directamente con el *home agent* o lo hará a través del *foreign agent*.
- **Túneles**: Para que los datagramas puedan ser entregados al *mobile node* cuando éste no se encuentra en su *home network*, el *home agent* debe crear un túnel hacia su *care-of address* para enviarle todos los datagramas dirigidos hacia él.

Con todos estos conceptos anteriores ya podemos describir el modo de funcionamiento habitual de Mobile IP:

1. Los *home agents* y *foreign agents* advierten su presencia mediante envíos de mensajes de aviso. En cualquier caso, un *mobile node* puede solicitar información sobre los agentes disponibles mediante el envío de mensajes de solicitud.
2. Un nodo móvil recibe un mensaje de aviso, y comprueba si está en su *home network* o no.
3. Cuando un nodo móvil detecta que se encuentra en su *home network*, éste funciona sin utilizar los servicios de movilidad. Si el nodo vuelve a su *home network* desde otro punto de conexión, des-registra su anterior posición en el *home agent*
4. Cuando un nodo móvil detecta que se ha movido a una *foreign network*, obtiene una *care-of address* de esa red.
5. El nodo móvil, si se encuentra fuera de su *home network*, registra su nueva *care-of address* en el *home network* para informar de su actual punto de acceso a la red.
6. Todos los datagramas enviados a la *home address* del nodo en su *home network* son interceptados por el *home agent* y enviados a través de un túnel. Estos datagramas son entregados en el otro extremo del túnel al *mobile node*.
7. En el otro sentido, los datagramas pueden ser enviados por el *mobile node* utilizando mecanismos de encaminamiento en IP tradicionales, sin necesidad de su paso por el *home agent*.

3.3. Cellular IP

3.3.1. Fundamentos de Cellular IP

Como hemos explicado en el apartado 3.2, las bases de los protocolos de micro/macro movilidad son que los paquetes dirigidos a un nodo móvil son entregados usando encaminamiento IP tradicional hacia la dirección asociada a este nodo móvil dependiendo del punto de conexión a la red. Este enfoque proporciona una solución simple y escalable para proporcionar una movilidad global. Pero Mobile IP no es apropiado, para ofrecer soluciones en entornos en los que el cambio de posición de un nodo es muy rápido, ya que en cada cambio el nodo debe obtener una dirección de la *foreign network* y debe comunicárselo a su, probablemente lejano, *home agent*.

Sin embargo, los sistemas de telefonía tradicionales están basados en un concepto diferente al de Mobile IP. En lugar de ofrecer un servicio de movilidad global, los sistemas telefónicos están optimizados para ofrecer handoffs A rápidos y suaves dentro de determinadas áreas geográficas restringidas.

Incluso en áreas geográficas reducidas, el número de usuario puede crecer hasta un punto en el que hacer búsquedas sobre la situación actual de cada nodo puede no ser viable. Además, la gestión de la movilidad de Mobile IP requiere que los nodos móviles manden información actualizada sobre su posición a su *home agent* cuando cambien de lugar, lo que significa una sobrecarga en el ancho de banda en las redes inalámbricas. Para solucionar este problema, las redes de telefonía, lo que hacen es que los nodos móviles solo se deben registrar después de un cambio de posición cuando estén teniendo transferencias o comunicaciones activas, en caso contrario, los nodos libres de comunicaciones mandan mensajes de información menos frecuentemente, con lo que pueden cambiar rápidamente de situación sin sobrecargar el sistema de gestión de la movilidad. De este modo, la situación de estos nodos libres solo es conocida aproximadamente, y si se necesita establecer una conexión con él solo se debe realizar su búsqueda por un limitado número de estaciones base.

Las redes de telefonía ofrecen una serie de características que si fueran aplicadas correctamente a las redes IP inalámbricas, podrían aumentar enormemente el rendimiento de estas redes sin perder la escalabilidad, flexibilidad y robustez que caracteriza a las redes IP. Pero la aplicación de estas características no es trivial, porque hay diferencias estructurales fundamentales entre los dos tipos de redes: los sistemas de telefonía requieren un modelo de encaminamiento por circuitos en los que debe crearse un camino de comunicación previo a realizarse las comunicaciones, y las redes IP poseen encaminamiento basado en paquetes.

Por todo ello, es necesaria la utilización de un protocolo como Cellular IP, para adaptar las características de las redes de telefonía a las redes IP inalámbricas.

3.3.2. Descripción del protocolo

Características

Cellular IP hereda los principios de los sistemas de telefonía para gestionar la movilidad, las conexiones pasivas y el control de handoff, pero adaptado a las redes IP.

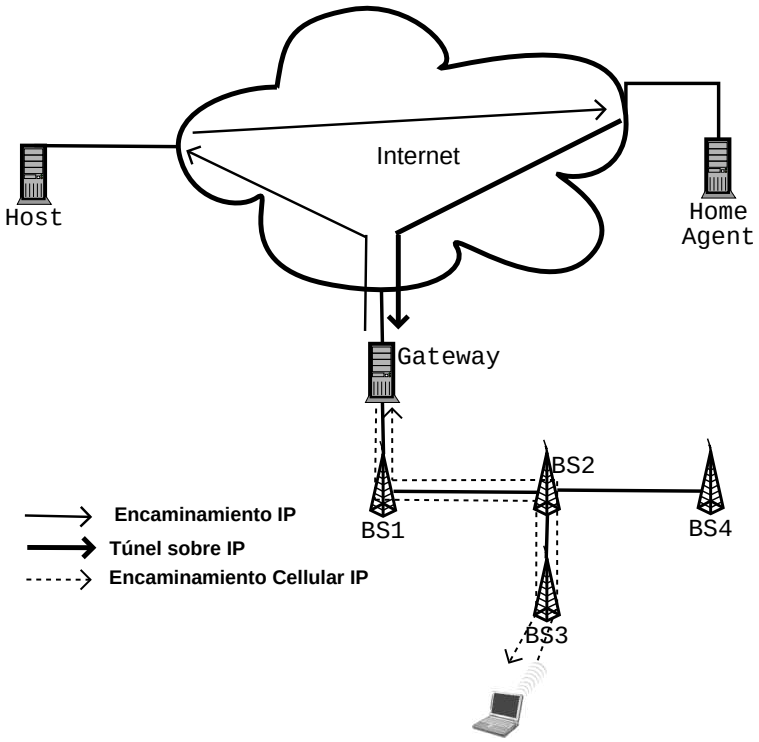


Figura 3.2: Ejemplo de red Cellular IP

El componente básico de una red Cellular IP es la **Estación Base**, el cuál sirve de punto de conexión a la red inalámbrica para los nodos pero también encamina los paquetes IP e integra el control de la funcionalidad de movilidad que proporciona el protocolo. Las estaciones base se basan en el encaminamiento, pero reemplazan el encaminamiento IP por el encaminamiento Cellular IP y gestionan la localización de nodos.

Una red Cellular IP está conectada a Internet mediante un **Gateway**, que es otro de los componentes básicos de estas redes.

La movilidad entre diferentes *gateways*, es decir, entre diferentes redes Cellular IP es gestionada por Mobile IP, mientras que la movilidad dentro de las áreas de los *gateways* es gestionada por Cellular IP. Los nodos móviles dentro de las redes Cellular IP usan la dirección del *gateway* como su dirección *care-of address* de Mobile IP.

En el caso general, los paquetes dirigidos al nodo móvil son recogidos por por el *home agent*, enviados a través de un túnel hasta el *gateway*, y este los encamina hacia las estaciones base. Dentro de las redes Cellular IP, los nodos móviles son identificados por su *home address* y los paquetes de datos son entregados desde las estaciones base sin necesidad de túneles ni conversión de direcciones. Los datos transmitidos desde el nodo móvil primero son encaminados hasta el *gateway* y después hacia Internet. Se puede observar un ejemplo en la figura 3.2.

La gestión de localización y el soporte para handoffs está integrado en el encaminamiento. Para minimizar los mensajes de control, los paquetes de datos normales transmitidos por los nodos móviles son usados para establecer la

localización de los mismos. Los paquetes transferidos desde el nodo móvil hasta el gateway son encaminados paso a paso por las estaciones base, y éstas apuntan en tablas caché el camino para llegar a estos nodos móviles. Cuando llegan paquetes hacia el nodo móvil las estación base usan esa información que tienen guardada en las tablas caché para entregarle los paquetes.

Cuando un nodo móvil no transmite datos durante un periodo de tiempo, envía datagramas IP vacíos al *gateway* para mantener activas las rutas en las estaciones base. Cuando un nodo ha estado inactivo durante un largo periodo de tiempo, las rutas hacia ese nodo son borradas de las tablas caché, y para poder encaminar paquetes de nuevo hacia ese nodo hace falta usar un mecanismo llamado *paginación*.

Encaminamiento

El *gateway* de una red Cellular IP periódicamente envía mediante broadcast mensajes para informar de su presencia. Las estaciones base, cuando reciben estos mensajes, registran el interfaz por el que lo han recibido, y lo usan para encaminar los paquetes hacia él. Todos los paquetes transmitidos desde los nodos móviles independientemente de su destino siempre son encaminados hacia el *gateway* usando estas rutas.

Según van pasando estos paquetes por cada uno de los nodos intermedios hacia el *gateway*, la información de encaminamiento es almacenada de la siguiente forma:

- Cada estación base mantiene una tabla caché de encaminamiento
- Cuando un paquete de datos originado por un nodo móvil entra en una estación base, ésta almacena la dirección IP del nodo móvil y el interfaz de red por el que le ha llegado el paquete.
- Esta información permanece válida durante un periodo de tiempo (*route-timeout*) y es actualizada cada vez que un paquete de datos entra por el mismo interfaz con el mismo nodo móvil de origen.
- Hay veces en las que un nodo quiere que la información de estas tablas no sea borrada aunque se haya cumplido el periodo de tiempo máximo, como por ejemplo un nodo que sea receptor de una conexión UDP en la que él no envía paquetes de datos, solo los recibe. Para conseguir esto, el nodo móvil debe enviar paquetes de actualización de rutas (*route-update-packets*) cada cierto tiempo (*route-update-time*).

Handoff

En Cellular IP existen dos algoritmos para la realización de los handoffs:

- *hard handoff*: Este algoritmo se basa en un enfoque simple de la gestión de la movilidad para dar soporte a rápidos y sencillos handoffs, pero con el precio de la potencial pérdida de paquetes.

  El handoff es iniciado por el nodo móvil. Éste recibe mensajes (*beacons*) de las estaciones base, y realiza un handoff en función de la intensidad de la señal de los mensajes que recibe de cada estación base. Cuando

el nodo decide hacer un handoff, se conecta a la nueva estación base y le manda un paquete para actualizar las rutas (*route-update*). Ésto va creando nuevas entradas en las tablas caché de las estaciones base hasta toda la información se ha actualizado. Durante el tiempo de latencia del handoff, los paquetes dirigidos al nodo pueden perderse porque hay rutas que no están actualizadas.

- *semisoft handoff*: Hay un periodo de tiempo en el que las rutas antiguas no se han borrado, porque no ha expirado el periodo de tiempo necesario, y durante este tiempo los datos son entregados desde las dos estaciones base, la nueva y la antigua. Gracias a esta característica este algoritmo consigue mejorar el rendimiento de los handoffs.

  Este algoritmo tiene dos partes:

  - Para reducir la latencia del handoff, las nuevas rutas deben ser creadas antes de que se produzca el cambio real del nodo a la nueva estación. Para ello, cuando el nodo móvil inicia el handoff, éste manda un paquete (*semisoft packet*) a la nueva estación base y vuelve a la antigua para seguir recibiendo paquetes. Mientras que continúa conectado a la antigua estación base, las rutas se van actualizando hacia la nueva. Después de que se han actualizado, ya puede cambiarse a la nueva estación base.
  - Durante el tiempo que las dos estaciones envían los datos al nodo, estos datos no están sincronizados, y por tanto podría haber problemas porque la estación nueva fuera más rápida que la antigua, y entonces perdería algunos paquetes al hacer el cambio real. Para solucionarlo, el paquete que envió el nodo por la nueva estación al comienzo del handoff, lo que hace es introducir un retardo en los paquetes que van por esa nueva ruta. De esta forma, nos aseguramos que no se van a perder paquetes, en todo caso podremos recibirlos repetidos, pero eso no es problema.

Paginación

Un nodo inactivo en Cellular IP es aquel que no ha recibido paquetes de datos durante un determinado periodo de tiempo (*active-state-timeout*), y después de cumplirse ese tiempo las entradas de ese nodo en las tablas caché de encaminamiento de las estaciones base son borradas.

Estos nodos inactivos envían paquetes (*paging-update*) cada determinado tiempo (*paging-update-time*) a la estación base de la que reciben una mejor calidad de señal. Estos paquetes son enviados hasta el *gateway*, y las estaciones base pueden almacenar entradas con esta información en las tablas caché de paginación.

Una tabla caché de paginación tiene el mismo formato y funciones que una tabla caché de encaminamiento, excepto por dos diferencias: el periodo de tiempo que permanecen las entradas en las tablas de paginación es mayor que en las tablas de encaminamiento, y las entradas de las tablas de paginación son actualización con cualquier paquete que sea enviado por el nodo móvil. Con este modo de funcionamiento, tenemos que los nodos inactivos tienen entradas en las tablas de paginación, pero no en las tablas de encaminamiento.

La **paginación** ocurre cuando un paquete tiene que ser enviado a un nodo y el *gateway* o las estaciones base no disponen de rutas de encaminamiento válidas hacia ese nodo:

- Si la estación no tiene tabla caché de paginación, entonces reenvía ese paquete por todos sus interfaces de red, exceptuando por el que lo recibió.
- Si la estación base tiene tabla caché de paginación, entonces solo reenvía el paquete si el nodo tiene una entrada válida en la tabla, y solo lo reenvía por el interfaz que está en la entrada de la tabla.

Si no se usan tablas caché de paginación, el primer paquete con destino a un nodo inactivo es enviado por toda la red mediante broadcast, lo que genera una sobrecarga en la red. Si se usan tablas caché de paginación el número de mensajes enviados se reduce.

Cuando un nodo inactivo recibe el paquete, cambia su estado a activo mediante el envío de un paquete de actualización de rutas (*route-update*).

3.4. IPv4 vs IPv6

IPv6 es un protocolo que ha sido desarrollado por el IETF, y pretende ser el reemplazo de la actual versión del protocolo IP (IPv4). De momento todavía no ha sido implantado como el protocolo a usar actualmente, pero está previsto que en un futuro cercano sea el protocolo del nivel de red usado por todas las máquinas en Internet.

Las principales restricciones que el protocolo IPv4 impone al actual crecimiento de Internet son:

- Número limitado de direcciones IP, 2^{32}
- Dificultad para manejar las tablas de encaminamiento

Pero IPv6 introduce muchas más mejoras con respecto a IPv4, y soluciona muchos problemas que existen en la actualidad. Y como no, en IPv6 el soporte para la movilidad, que es el tema que nos interesa, es mucho más útil, y también ha sido diseñado teniendo en cuenta estos objetivos.

Las principales diferencias entre Mobile IPv4 y Mobile IPv6 son:

- Lo que se conocía en Mobile IPv4 como *Route optimization* ahora forma parte del protocolo en Mobile IPv6. Esto significa que los paquetes dirigidos desde el *correspondent node* hasta el nodo móvil lo hacen directamente y no tienen que pasar por el *home agent* y luego ser reenviados hasta el nodo móvil.
- En este protocolo los paquetes que manda el nodo móvil llevan como dirección origen la *care-of address* en la cabecera IP, y luego llevan una opción para el destino con la dirección *home address*. Esto, a diferencia de Mobile IPv4, hace que sea transparente para todos los encaminadores y para las capas superiores.
- El uso de la *care-of address* como la dirección origen de los paquetes IP simplifica el encaminamiento de paquetes multicast enviados por el nodo móvil. En Mobile IPv4, el nodo móvil tenía que hacer un túnel hasta su *home agent* para poder usar de forma transparente su *home address* como dirección origen de los paquetes multicast. En Mobile IPv6, con la opción de destino de la *home address* permite ser compatible con el encaminamiento multicast, que en parte está basado en la dirección origen del paquete.
- Ya no hace falta tener encaminadores especiales que hagan de *foreign agent* como en Mobile IPv4. Ahora, el nodo móvil usa las características que le proporciona IPv6, tales como auto-configuración de dirección y *neighbor discovery*.
- La mayoría de los paquetes que se envían a un nodo móvil cuando no está en la *home network* se hace usando una cabecera de encaminamiento IPv6 (*IPv6 Routing Header*) en lugar de encapsulación de IP (como en Mobile IPv4), y esto hace que se reduzcan los bytes necesarios en la cabecera.

- Cuando un nodo móvil no está en la *home network*, el *home agent* intercepta cualquier paquete que se dirige hacia el nodo móvil usando *IPv6 Neighbor Discovery* en lugar de ARP como en Mobile IPv4. Esto simplifica la implementación de Mobile IP al ser independiente de la capa de enlace, cosa que no ocurre con ARP.
- El mecanismo de descubrimiento de la dirección del *home agent* dinámica en Mobile IPv6 usa IPv6 anycast y devuelve una sola respuesta al nodo móvil, al contrario de Mobile IPv4 que usaba mensajes de broadcast y una respuesta de cada uno de los *home agents*. El mecanismo de Mobile IPv6 es más eficiente y más seguro.

3.5. Prácticas sobre protocolos de movilidad en IPv4

En esta sección se van a describir todos los detalles de las pruebas realizadas sobre varias implementaciones de protocolos de micro/macro movilidad sobre IPv4:

- Implementación Dynamics de Mobile IPv4, de la Helsinki University of Technology [HUT99]
- Implementación de Cellular IPv4 desarrollada por la universidad de Columbia [Uni99]

Para las pruebas de estos protocolos se ha construido una maqueta de ordenadores, con un diseño de red específico para cada una de las pruebas. La parte fundamental de la maqueta se basa en el esquema de la figura 3.3.

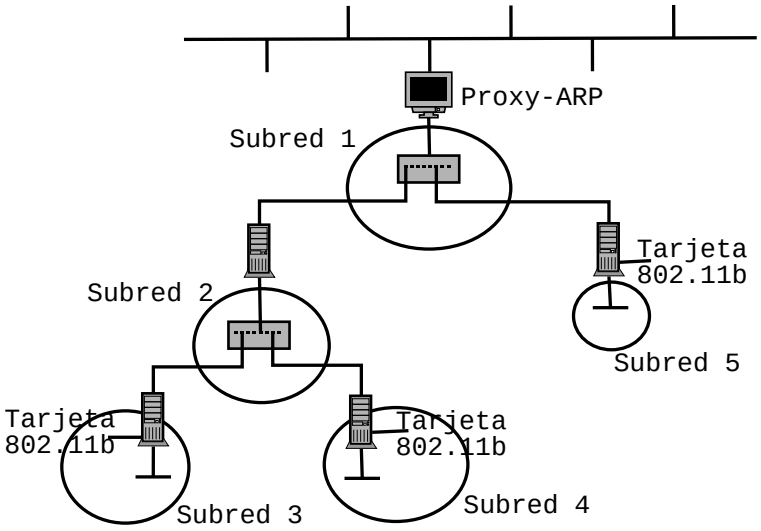


Figura 3.3: Maqueta para pruebas de movilidad en IPv4

El objetivo de este esquema es poder tener el mayor número de subredes, con el menor número de ordenadores. Tenemos 5 ordenadores, y también tenemos 5 subredes. Solamente 1 ordenador está conectado directamente a Internet, y hace de ProxyARP para que los demás también puedan acceder.

Las direcciones IP que usamos para las subredes tienen la siguiente estructura: 29 bits para la dirección de red, y 3 bits para la dirección de máquina. Por tanto podemos tener 6 máquinas en cada una de las subredes, lo que es más que suficiente ..

3.5.1. Montaje de Mobile IPv4 en la maqueta

La implementación de Mobile IP utilizada para el montaje sobre la maqueta es la de la universidad de Helsinki [HUT99].

Para la instalación de Mobile IP sobre la maqueta la configuramos de la siguiente manera:

Tenemos las 5 subredes:

- Subred 1: 193.147.71.40 / 255.255.255.248
- Subred 2: 193.147.71.8 / 255.255.255.248
- Subred 3: 193.147.71.16 / 255.255.255.248
- Subred 4: 193.147.71.24 / 255.255.255.248
- Subred 5: 193.147.71.32 / 255.255.255.248

Tenemos un nodo móvil, con dirección 212.128.1.104, tiene una tarjeta inalámbrica 802.11b mediante la que se conecta a los *foreign agents* (FA3, FA4 y FA2) que también tienen tarjetas 802.11b, y todas ellas están configuradas de modo Ad-Hoc.

El nodo móvil se va moviendo por las subredes, y va cambiando de *foreign agent* sin cambiar su dirección IP, ya que usamos *foreign agent* encapsulation. De este modo los paquetes que van desde el *correspondent node* al nodo móvil pasan por el *home agent*, que encapsula los paquetes y se los envía al *foreign agent* mediante un túnel, y después es el *foreign agent* el que des-encapsula el paquete y se lo entrega al nodo móvil.

Con la maqueta que tenemos también hacemos uso de los *foreign agents* jerárquicos. Esto ocurre entre el FA1 y los FA3 y FA4. Esto proporciona cambios de subred para el nodo móvil más rápidos, ya que el túnel que tiene el *home agent* siempre lo tiene con el FA1, y después dependiendo de si el nodo móvil está en el FA3 o el FA4, existirá otro túnel entre el FA1 y el FA3, o entre el FA1 y el FA4. Por eso cuando el nodo móvil cambia por ejemplo del FA3 al FA4, lo único que cambia es que el túnel entre FA1-FA3 se elimina y se crea otro entre FA1-FA4.

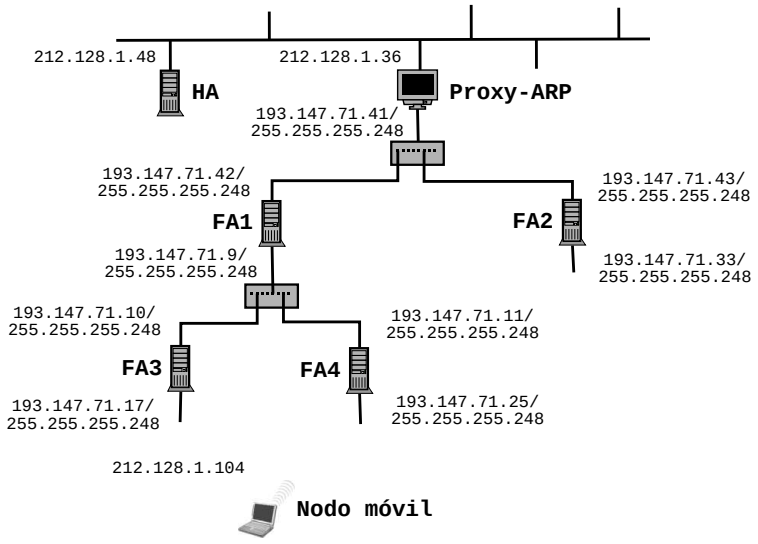


Figura 3.4: Diseño de red para pruebas de Mobile IPv4

En el caso de no usar *foreign agents* jerárquicos, como por ejemplo al cambiar del FA3 al FA2, el establecimiento de conexión del nodo móvil tiene un mayor retardo, ya que se debe establecer un túnel entre el *home agent* y el FA2 y eso es más lento que el caso que se comentaba antes.

3.5.2. Montaje de Cellular IPv4 en la maqueta

La implementación de Cellular IP utilizada es la de la Universidad de Columbia [Uni99].

La configuración de la maqueta para el uso de Cellular IP es que se muestra en la figura 3.5.

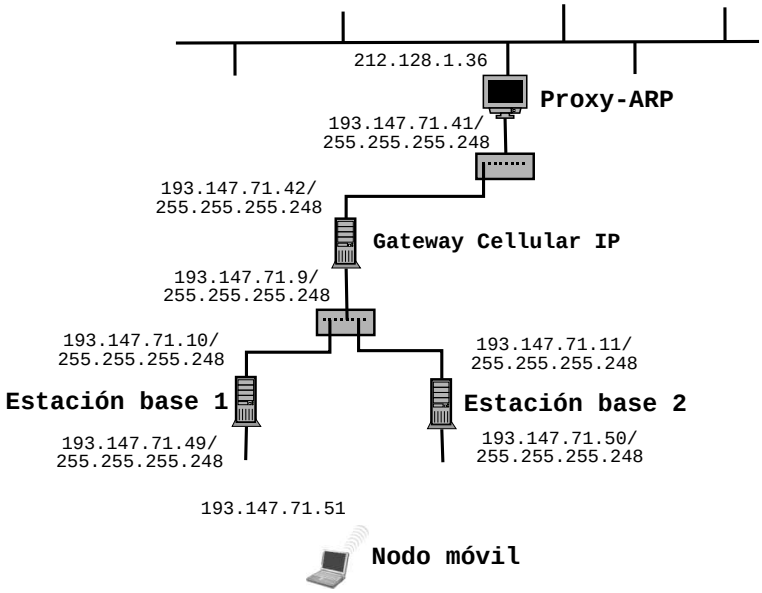


Figura 3.5: Diseño de red para pruebas de Cellular IPv4

Con Cellular IP, el nodo móvil pertenece a las subredes de las estaciones base para poder conectarse a ellos.

En este caso el nodo móvil tiene una tarjeta inalámbrica, al igual que las estaciones base. Todas estas tarjetas están configuradas en modo ad-hoc.

Con Cellular IP, en todo momento el *gateway* sabe a que estación base está conectado el nodo móvil, y lo que hace es encaminar los paquetes desde/hacia el nodo móvil por la estación base adecuada en cada momento.

Los cambios de estación base son más rápidos que los cambios de *foreign agent* en Mobile IP, ya que aquí solo hay que cambiar una ruta en el *gateway* de CellularIP y no hace falta establecer túneles ni encapsular paquetes que luego hay que des-encapsular.

3.6. Medida de prestaciones de Mobile IPv4

Una vez que tenemos instalada la implementación de Mobile IPv4 sobre la maqueta como comentamos en la sección 3.5.1, hemos realizado unas pruebas para medir el rendimiento de la implementación utilizada.

Hemos realizado dos tipos de pruebas, que describimos en las siguientes secciones, así como las herramientas utilizadas.

3.6.1. Herramientas utilizadas para las pruebas de Mobile IPv4

Las pruebas sobre Mobile IP en la maqueta se han realizado con algunas herramientas existentes para este fin, y otras herramientas implementadas para algunas pruebas. Para automatizar las pruebas se han realizado scripts en perl. En estos scripts se define el tipo de pruebas, se lanza el proceso, y se generan algunos ficheros con los resultados.

Básicamente existe un scripts principal en el que se define el tipo de pruebas, la duración, el número de repeticiones, el programa utilizado para realizar la medición, ... Y luego existen tantos scripts como herramientas de medición existen, en el que se indica la forma de ejecutar el programa de medición, y la forma de recoger sus resultados. Los programas utilizados para las mediciones son netperf, y una implementación de un programa cliente/servidor que básicamente consiste en la medida de los paquetes que se pierden en una transferencia de paquetes entre el cliente/servidor.

3.6.2. Pruebas ancho de banda en Mobile IPv4

Estas pruebas realizadas consisten en medir el ancho de banda que se consigue entre el nodo móvil (MN) y el *Correspondent Node* (CN), sobre distintos escenarios.

Para las pruebas se ha utilizado la herramienta netperf. Esta herramienta consiste en un modelo cliente/servidor, ejecutamos el servidor en el CN y el cliente en el MN. Durante el tiempo que dura cada prueba este programa realiza una transferencia entre ambos y mide el ancho de banda que se consigue...

Para la automatización de las pruebas se han construido unos scripts en perl, que realizan todas las pruebas que se le indiquen durante el tiempo que se le indique.

Las pruebas con netperf consisten en medir el ancho de banda conseguido entre ambos nodos, cuando el número de handoffs del MN varía. Se han tomado medidas cuando el MN no hace handoffs, y también se han hecho medidas cuando el MN hace unos handoffs de 8, 5, 4, 3 y 2 segundos. Los handoffs consisten en que el MN va cambiando de FA al que se conecta.

También se han hecho pruebas utilizando FA jerárquicos y sin utilizarlos para comprobar cual es la mejora que proporcionan los FA jerárquicos, en el que el túnel entre el HA y el FA superior no varía.

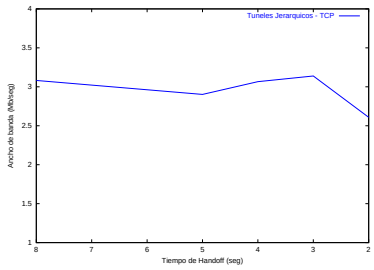
También se ha utilizado el programa nistnet, que sirve para variar el comportamiento de ciertos paquetes en la red. Por ejemplo para nuestras pruebas hemos hecho que el nistnet retrase todos los paquetes desde/hacia el HA 1 segundo, para simular cual sería la situación en la que el HA se encontrara lejos del MN.

Resultados con FA jerárquicos

- **Sobre TCP**

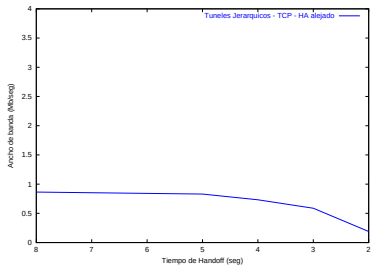
  Ancho de banda:

Handoffs (seg.)	10	5	4	3	2
Ancho de banda (Mb/seg)	3.20125	2.9025	3.06625	3.13875	2.8025



  Ancho de banda alejando al HA:

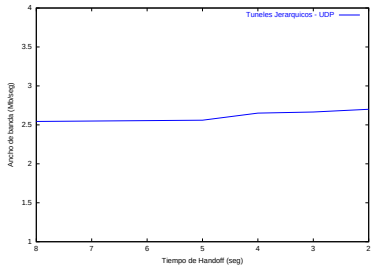
Handoffs (seg.)	10	5	4	3	2
Ancho de banda (Mb/seg)	0.8875	0.83	0.7325	0.5875	0.19



- **Sobre UDP**

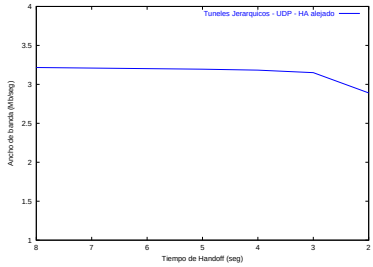
  Ancho de banda:

Handoffs (seg.)	10	5	4	3	2
Ancho de banda (Mb/seg)	2.5325	2.56	2.65125	2.665	2.7



Ancho de banda alejando al HA:

Handoffs (seg.)	10	5	4	3	2
Ancho de banda (Mb/seg)	3.23	3.195	3.1825	3.15	2.89

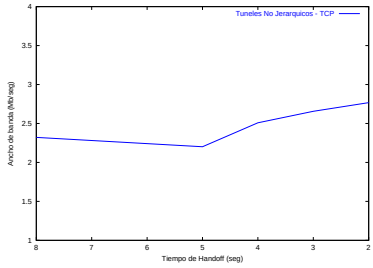


Resultados sin FA jerárquicos

- **Sobre TCP**

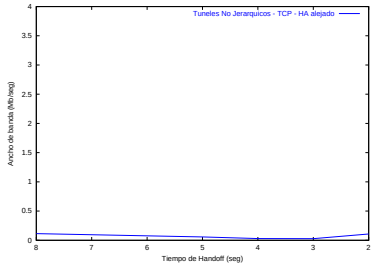
Ancho de banda:

Handoffs (seg.)	10	5	4	3	2
Ancho de banda (Mb/seg)	2.40125	2.20125	2.50875	2.65625	2.7675



Ancho de banda alejando al HA:

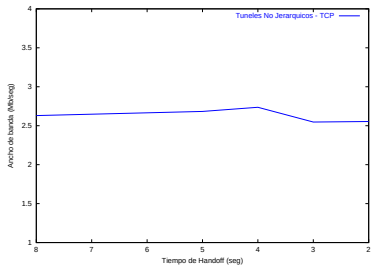
Handoffs (seg.)	10	5	4	3	2
Ancho de banda (Mb/seg)	0.1525	0.0575	0.03	0.0275	0.1075



- **Sobre UDP**

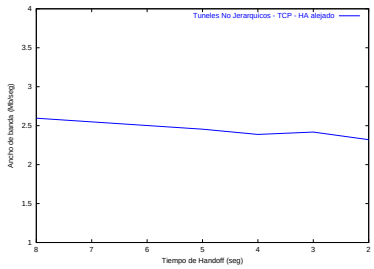
Ancho de banda:

Handoffs (seg.)	10	5	4	3	2
Ancho de banda (Mb/seg)	2.595	2.683	2.73625	2.5475	2.5537



Ancho de banda alejando al HA:

Handoffs (seg.)	10	5	4	3	2
Ancho de banda (Mb/seg)	2.69	2.455	2.3875	2.4175	2.32



3.6.3. Pruebas pérdida de paquetes UDP en Mobile IP

Otro tipo de pruebas realizado consiste en medir el número de paquetes que se pierden entre el MN y el CN, con diferentes parámetros de configuración de la maqueta.

Para las pruebas de pérdida de paquetes se han realizado unos simples programas en Ada, utilizando la librería de comunicaciones Lower Layer, que consisten en un modelo cliente/servidor en el que se envían paquetes UDP de distinto tamaño, y se mide el número de paquetes perdidos.

En las pruebas de pérdidas de paquetes el método es el mismo pero se han realizado con tiempos de handoffs de 15, 12, 10, 8, 6, 5, 4, 3 y 2 segundos.

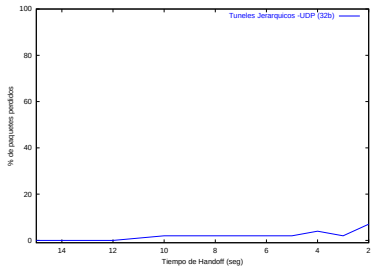
También se han usado FA jerárquicos y FA no jerárquicos, y también se ha utilizado la herramienta nistnet para alejar al HA.

Resultados con FA jerárquicos

- Con tamaño de paquete de 32 bytes

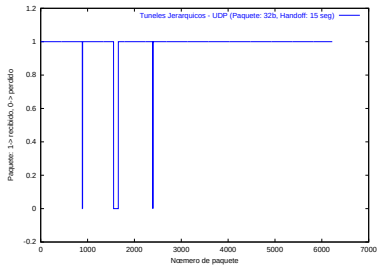
  Pérdida de paquetes:

Handoffs (seg.)	15	12	10	8	6	5	4	3	2
Paquetes perdidos (%)	0	0	2	2	2	2	4	2	7

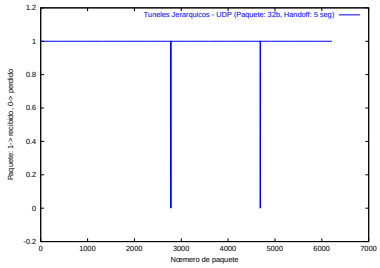


Muestra de los mensajes que se pierden en el tiempo que duran las pruebas con alguno de los handoffs:

- Con handoffs de 15 segundos:

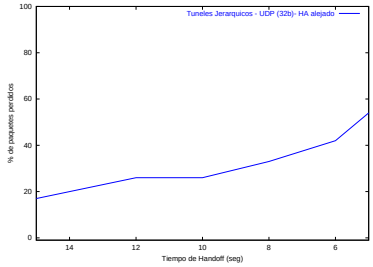


- Con handoffs de 5 segundos:



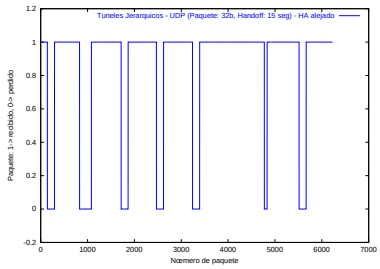
Pérdida de paquetes alejando al HA:

Handoffs (seg.)	15	12	10	8	6	5
Paquetes perdidos (%)	17	26	26	33	42	54

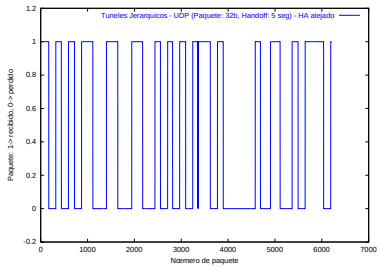


Muestra de los mensajes que se pierden en el tiempo que duran las pruebas con alguno de los handoffs:

- Con handoffs de 15 segundos:



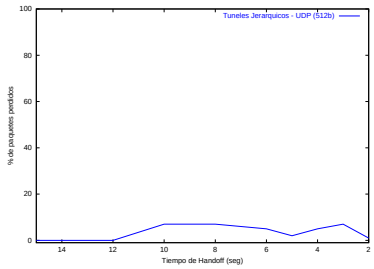
- Con handoffs de 5 segundos:



- Con tamaño de paquete de 512 bytes

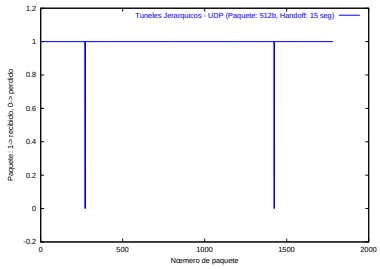
  Pérdida de paquetes:

Handoffs (seg.)	15	12	10	8	6	5	4	3	2
Paquetes perdidos (%)	0	0	7	7	5	2	5	7	1

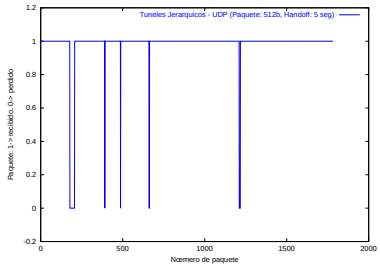


Muestra de los mensajes que se pierden en el tiempo que duran las pruebas con alguno de los handoffs:

- Con handoffs de 15 segundos:

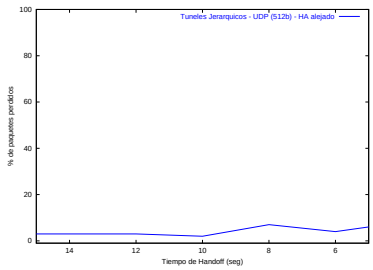


- Con handoffs de 5 segundos:



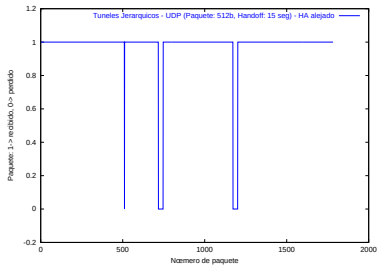
Pérdida de paquetes alejando al HA:

Handoffs (seg.)	15	12	10	8	6	5
Paquetes perdidos (%)	3	3	2	7	4	6

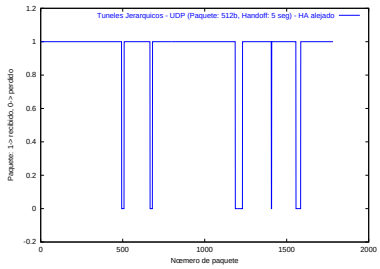


Muestra de los mensajes que se pierden en el tiempo que duran las pruebas con alguno de los handoffs:

- Con handoffs de 15 segundos:



- Con handoffs de 5 segundos:

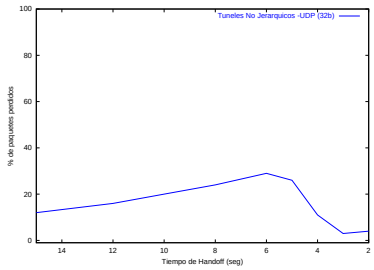


Resultados sin FA jerárquicos

- Con tamaño de paquete de 32 bytes

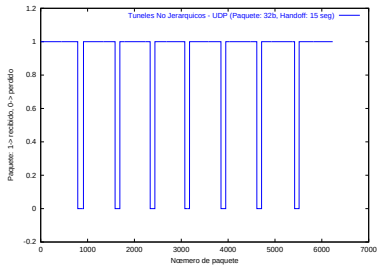
  Pérdida de paquetes:

Handoffs (seg.)	15	12	10	8	6	5	4	3	2
Paquetes perdidos (%)	12	16	20	24	29	26	11	3	4

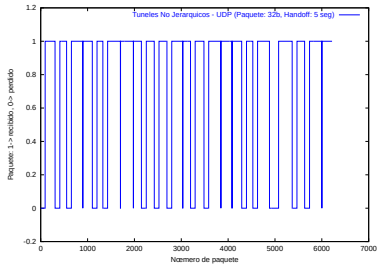


Muestra de los mensajes que se pierden en el tiempo que duran las pruebas con alguno de los handoffs:

- Con handoffs de 15 segundos:

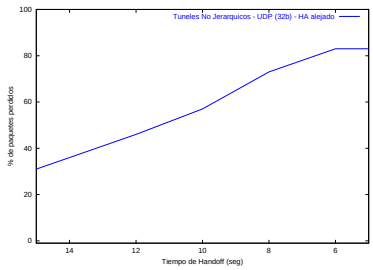


- Con handoffs de 5 segundos:



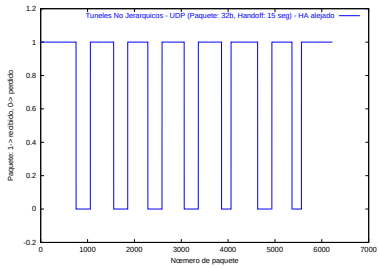
Pérdida de paquetes alejando al HA:

Handoffs (seg.)	15	12	10	8	6	5
Paquetes perdidos (%)	31	46	57	73	83	83

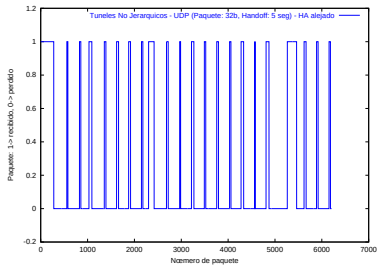


Muestra de los mensajes que se pierden en el tiempo que duran las pruebas con alguno de los handoffs:

- Con handoffs de 15 segundos:



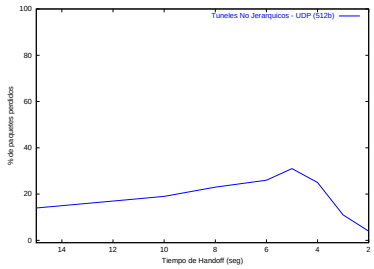
- Con handoffs de 5 segundos:



- Con tamaño de paquete de 512 bytes

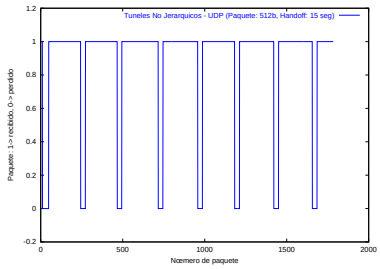
  Pérdida de paquetes:

Handoffs (seg.)	15	12	10	8	6	5	4	3	2
Paquetes perdidos (%)	14	17	19	23	26	31	25	11	4

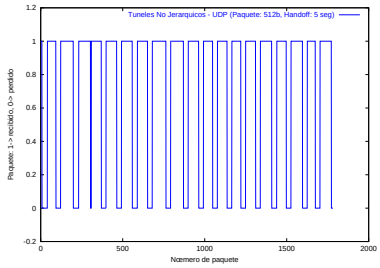


Muestra de los mensajes que se pierden en el tiempo que duran las pruebas con alguno de los handoffs:

- Con handoffs de 15 segundos:

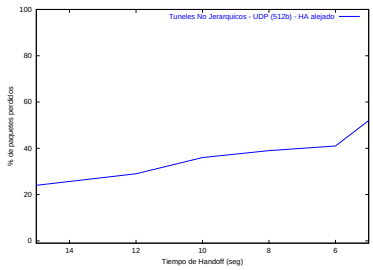


- Con handoffs de 5 segundos:



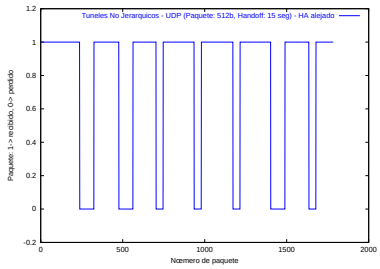
Pérdida de paquetes alejando al HA:

Handoffs (seg.)	15	12	10	8	6	5
Paquetes perdidos (%)	24	29	36	39	41	52

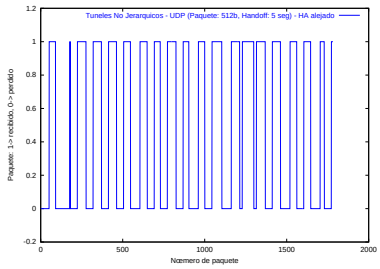


Muestra de los mensajes que se pierden en el tiempo que duran las pruebas con alguno de los handoffs:

- Con handoffs de 15 segundos:



- Con handoffs de 5 segundos:



3.7. Prácticas sobre protocolos de movilidad en IPv6

En esta sección se van a describir todos los detalles de las pruebas realizadas sobre varias implementaciones de protocolos de micro/macro movilidad sobre IPv6:

- Implementación MIPL de Mobile IPv6, de la Helsinki University of Technology [TL00]
- Implementación de Cellular IPv6 desarrollada por Sanghyo Kim y Javier Gomez [SK01]

Para las pruebas de estos protocolos se ha construido una maqueta de ordenadores, con un diseño de red específico para cada una de las pruebas.

Básicamente la maqueta 3.6 consiste en un conjunto de ordenadores a los que se conecta el nodo móvil dependiendo de su localización actual, y el objetivo es que este nodo pueda seguir teniendo conectividad independientemente del nodo al que se conecte y de manera transparente para el usuario.

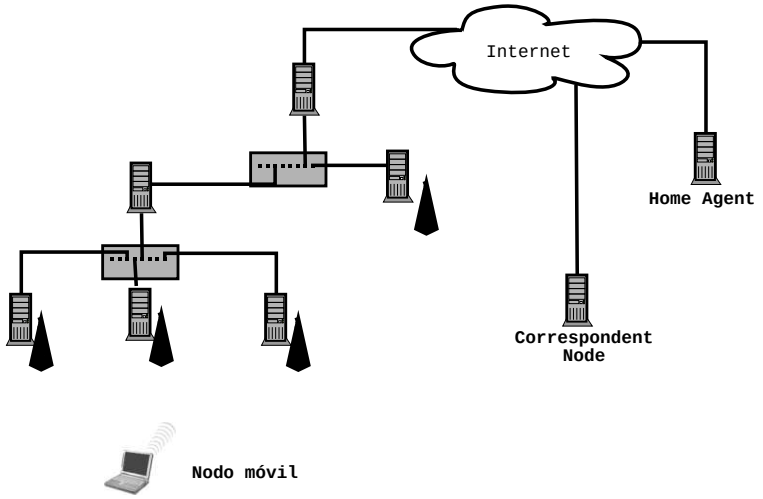


Figura 3.6: Maqueta para pruebas de movilidad en IPv6

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

Primero detallaré el proceso de montaje de la maqueta para que funcione en IPv6, y después explicaré la instalación de las implementaciones de Mobile IPv6 y Cellular IPv6.

3.7.1. Montaje de la maqueta con IPv6

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

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

Configuración del kernel con soporte para IPv6

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

Entre las características que proporciona el módulo están:

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

Para tener soporte para IPv6 y para la posterior instalación de las implementación de los protocolos de movilidad necesitamos tener compilado un kernel que nos facilite los siguientes módulos:

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

Configuración de las tarjetas PCMCIA 802.11b

Las tarjetas utilizadas en la maqueta son varias Lucent Technologies, y algunas Compaq pero que también tienen el chip de Lucent. Se utilizarán en modo ad-hoc (sección 2.8) porque no disponemos de Puntos de Acceso y necesitamos que las tarjetas puedan comunicarse entre ellas directamente.

La configuración de las tarjetas se realiza como se detalla en la sección 2.4.4, más las siguientes modificaciones que se realizan para configurar el modo de funcionamiento por defecto.

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

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

Diseño de la maqueta

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

Con la maqueta que hemos diseñado tenemos 6 subredes distintas 3.7, lo que nos permitirá posteriormente bastantes posibilidades para desarrollar nuestras pruebas:

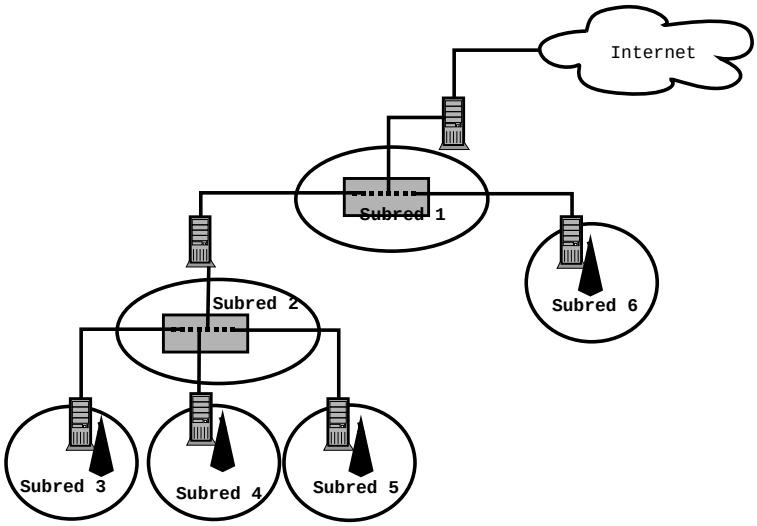


Figura 3.7: Diseño de red de la maqueta IPv6

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

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

- **Subred 6:** fec0:0:0:6::/64

Configuración de las direcciones IPv6

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

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

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

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

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

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

Cuando una máquina que se conecta a esa subred recibe un mensaje de ese tipo se configura automáticamente una dirección adecuada para esa subred. Y para ello lo que hace es coger la parte de dirección de red que recibe y

después le adjunta la dirección MAC de su tarjeta. Por ejemplo una máquina con dirección MAC 00:50:DA:4F:A7:87 obtiene la siguiente dirección en esa subred: fec0::1:250:daff:fe4f:a787.

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

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

Configuración de las rutas IPv6

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

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

3.7.2. Instalación de la implementación de Mobile IPv6

El primer paso es bajarse las fuentes del kernel y dejarlas en un lugar habitual para compilarlo posteriormente.

A continuación hay que descargarse la versión de Mobile IPv6 de la implementación que estamos usando [TL00], dependiendo de la versión del kernel que nos hayamos bajado anteriormente (2.4.7). Este software descargado contiene un parche que hay que aplicar a los fuentes del kernel. Para ello se copia el parche (`mipv6-0.9-v2.4.7.patch`) al directorio donde están esos fuentes y se aplica el parche de la siguiente forma:

```
# patch -p1 < mipv6-0.9-v2.4.7.patch
```

Con esto ya están modificados los fuentes del kernel para soportar Mobile IPv6, y lo que hay que hacer después es compilar las fuentes del kernel con soporte para los siguientes módulos:

- IPv6: Mobility Support (EXPERIMENTAL) (m)
- MIPv6: Debug Messages (m)

Compilamos el kernel, y lo instalamos de la manera habitual en la distribución que usamos con la herramienta *make-kpkg*.

Después de esto lo único que falta es crear un dispositivo que necesita la implementación de Mobile IPv6:

```
#  mknod /dev/mipv6_dev c 0xf9 0
```

Configuración de la maqueta para Mobile IPv6

En cada uno de los nodos que realizan funciones de Mobile IP (nodo móvil, *home agent* y *foreign agents*) debemos tener instalados correctamente el módulo de Mobile IPv6. Para la configuración de dicho módulo existen varios ficheros de configuración:

- */etc/sysconfig/network-mip6.conf*: Fichero principal de configuración
- */etc/mipv6_acl.conf*: Lista de control de acceso de nodos móviles.
- */etc/mipv6_sas.conf*: Seguridad de Mobile IPv6.

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

- **Nodo móvil**. El fichero */etc/sysconfig/network-mip6.conf* contiene lo siguiente:

```
# MIPL Mobile IPv6 Configuration file

FUNCTIONALITY=mn
DEBUGLEVEL=7
# TUNNEL_SITELOCAL=yes
HOMEADDRESS=fec0::1:260:1dff:fef1:2be8/64
HOMEAGENT=fec0::1:250:4ff:fe47:d29a
# MOBILENODEFILE=/etc/mipv6_acl.conf
RTR_SOLICITATION_INTERVAL=1
RTR_SOLICITATION_MAX_SENDTIME=5
```

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

```
# MIPL Mobile IPv6 Configuration file

FUNCTIONALITY=ha
DEBUGLEVEL=1
TUNNEL_SITELOCAL=yes
# HOMEADDRESS=fec0::1:260:1dff:fef1:2be8
# HOMEAGENT=fec0::1:250:4ff:fe47:d29a
MOBILENODEFILE=/etc/mipv6_acl.conf
# RTR_SOLICITATION_INTERVAL=1
# RTR_SOLICITATION_MAX_SENDTIME=5
```

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

```
ALLOW fec0::1:260:1dff:fef1:2be8/64
```

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

```
# MIPL Mobile IPv6 Configuration file

FUNCTIONALITY=cn
```

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

Aparte de estos elementos necesitamos que los encaminadores de las subredes a las que se conecte el nodo móvil dispongan de un radvd ejecutando, como se explicó en la sección 3.7.1.

Funcionamiento de Mobile IPv6 en la maqueta

Una vez configurados todos los nodos, solamente debemos activar los módulos de Mobile IPv6. Para ello en cada una de las máquinas (nodo móvil, *home agent* y *correspondent node*) debemos hacer lo siguiente:

/etc/init.d/mobile-ip6 start

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

Si el nodo móvil se encuentra en la *home network* el funcionamiento de éste será el normal, podrá enviar y recibir paquetes como si fuera un nodo convencional.

Cuando el nodo móvil no se encuentra en la *home network* lo primero que hace es adquirir una nueva dirección de la subred a la que se conecta (*care-of address*). Después de esto necesita que su *home agent* conozca esa nueva dirección, y para ello le manda un *binding registration* (paquete con opción para el destino *Binding Update*), y el *home agent* le responde con una paquete con opción para el destino *binding acknowledgement*. A partir de ese momento esa nueva dirección será la *primary care-of address*, y el *home agent* interceptará los paquetes dirigidos hacia el nodo móvil mediante *proxy Neighbor Discovery*, y se los enviará mediante *IPv6 encapsulation*. Cada vez que el nodo móvil se cambie de subred mandará al *home agent* un *binding update*.

Podemos probar a conectar el nodo móvil a cualquiera de las subredes que tenemos y veremos que va adquiriendo nuevas direcciones y puede seguir comunicándose con el *correspondent node* mediante algún programa que tenga soporte para IPv6 (ssh, ping6, ...).

3.7.3. Instalación de la implementación de Cellular IPv6

La instalación de Cellular IPv6 es más sencilla que la de Mobile IPv6. La maqueta sobre la que se puede probar esta implementación debe tener una estructura jerárquica como la que se muestra en la figura 3.8

Los requerimientos del sistema son los siguientes:

- Kernel a partir de 2.2.12. Nosotros hemos utilizado el 2.4.7.
- Tener instalado el paquete *iproute*.
- Tener instaladas las librerías *libpcap* con soporte ipv6.

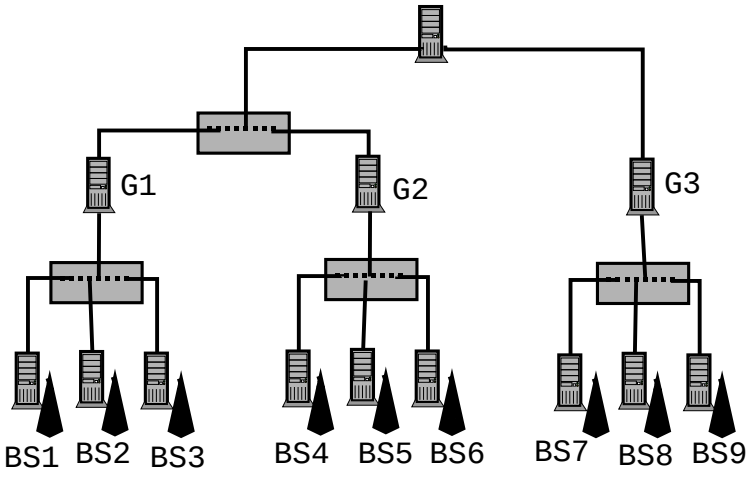


Figura 3.8: Maqueta jerárquica para pruebas sobre Cellular IPv6

- Se pueden usar tarjetas wireless que soporten funciones SPY, para medir la calidad de la señal.

Lo primero que hay que hacer es bajarse la implementación que vamos a usar [SK01].

La instalación es muy sencilla. Primero se descomprimen los fuentes. Una vez descomprimidos los fuentes vemos que se crean dos directorios: cipmobile6 y cipnode6. En el primero de ellos está el código que se ejecutará en el nodo móvil y en el segundo está el código que se ejecutará en todos los demás nodos, estaciones base y *gateway*. Después entramos en el directorio correspondiente, ejecutamos: make y ya se compilarán los fuentes necesarios.

Configuración de los nodos

La configuración de los nodos es la siguiente:

- En el nodo móvil:

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

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

- En las estaciones base:

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

- Estación base 1

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

- Estación base 2

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

- En el *gateway*:

  La configuración del *gateway* se especifica en el fichero *cipnode.conf*:

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

Funcionamiento de Cellular IPv6 en la maqueta

Como hemos dicho anteriormente, Cellular IP es un protocolo de micromovilidad, por lo tanto debería ir acompañado de un protocolo de macro-

movilidad como Mobile IP. Por tanto en nuestra maqueta se usa Mobile IPv6 junto con Cellular IPv6, pero para su correcto funcionamiento deberían interoperar perfectamente y las implementaciones que estamos probando de ambos protocolos no lo hacen.

Para que interoperen entre los dos hemos tenemos que hacer algunas operaciones extra al ponerlos en funcionamiento. Una de las razones por las que no interoperan correctamente es la siguiente: el *gateway* de Cellular IP emite unos paquetes (*Gateway Broadcast Packet*) que permiten a las estaciones base conocer el prefijo de subred del *gateway* y utilizar esta información a la hora de enviar los beacons que reciben los nodos móviles y que les permite autoconfigurar sus direcciones. Esto lo hace correctamente. Pero a la vez que ocurre ésto, MobileIP debería darse cuenta de esa nueva dirección y hacer que esa dirección fuera su *care-of address*, pero esto no ocurre así. Para solucionar esto lo que hago es poner en las estaciones base un radvd para que emitan el prefijo de subred del *gateway*.

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

Con esto ya conseguimos que el módulo de Mobile IP en el nodo móvil se entere de esa nueva dirección, la use como su *care-of address* y se registre en el *home agent* correctamente.

Los pasos a seguir para poner la maqueta en funcionamiento son los siguientes:

- Arrancar la maqueta de Cellular IP sin radvds en las estaciones base
- Arrancar Mobile IP en el nodo móvil y llevarlo a la *home network* para que adquiera la dirección *home address*.
- Llevarlo de nuevo a la *foreign network*.
- Arrancar los radvds en las estaciones base.
- Después de esto el nodo móvil debe registrarse correctamente en el *home agent*, y ya podemos hacer handoffs entre las estaciones base.

Capítulo 4

Computación ubicua

Mark Weiser, en Septiembre de 1991 [Wei91] describió su visión de lo que él llamaba computación ubicua, hoy llamada computación pervasiva. La esencia de su visión era la creación de entornos repletos de computación y de capacidad de comunicación, todo integrado de forma inapreciable junto a los personas. La visión de Weiser estaba bastante alejada de su época, entre otras razones porque no existía la tecnología necesaria para llevarlo a cabo.

Pero después de más de una década de progreso en el campo de los dispositivos hardware, las criticadas ideas de Weiser en 1991 ahora son productos comercialmente viables:

- Ordenadores de bolsillo
- Redes inalámbricas
- Sensores muy avanzados
- Computación "vestible"

Hoy en día existen muchos proyectos de investigación sobre computación pervasiva, tanto en las universidades como en las empresas: Oxygen en el MIT[MIT], Aura en el CMU[CMU], CoolTown en HP[HP], ... Cada uno de estos proyectos se centran en diferentes aspectos de la computación ubicua, y persiguen diferentes objetivos tanto a largo como a corto plazo, pero todos ellos intentan hacer de la computación pervasiva una realidad [Sat95].

4.1. Principios

Uno de los principales objetivos de la computación ubicua es hacer desaparecer a los dispositivos computacionales haciéndolos situarse en un segundo plano. Este objetivo de crear dispositivos que se mezclen en la vida cotidiana hasta que lleguen a ser indistinguibles supone una potencial revolución que puede hacer cambiar el modo de vida diario. Las personas se centrarán en las tareas que deben hacer, no en las herramientas que utilizan, porque se pretende que esas herramientas pasen desapercibidas.

El significado de enviar la computación a un "segundo plano", está referido a dos conceptos diferentes pero relacionados [Ara95]. El primero es el significado literal de que la tecnología de la computación se debe integrar en los objetos, cosas, tareas y entornos cotidianos. Y la segunda es que esta integración se debe realizar de forma que la introducción de computación en estas cosas u objetos no interfieran con las actividades para las que son usadas, y que siempre proporcionen un uso más cómodo, sencillo y útil de esos objetos.

Estos objetos cotidianos en los que se integra la tecnología de la computación pasan a tener una serie de propiedades que permiten la creación del entorno ubicuo buscado. Estas son algunas de esas propiedades:

- **Comunicación** entre dispositivos: todos estos objetos dotados de capacidad de computación también tienen capacidad de comunicación, y no solo con el usuario, sino con los demás objetos integrados que haya a su alrededor
- Estos objetos tienen **memoria**: Además de poder comunicarse entre ellos e interactuar con los usuarios, estos dispositivos tienen capacidad de memoria y pueden utilizar esta memoria para una mejor interacción con el resto de dispositivos.
- Son **sensibles al contexto**: estos objetos son sensibles al contexto, es decir, se adaptan a las posibles situaciones, como la situación geográfica, los dispositivos que hay a su alrededor, las preferencias de los usuarios, ... y actúan dependiendo de ese entorno que los rodea.
- Son **reactivos**: estos objetos reaccionan a ocurrir determinados eventos, que pueden percibir en su entorno mediante sensores o a través de la interacción con otros dispositivos.

4.2. Motivaciones para la computación ubicua

Existen algunos factores que hacen que la computación ubicua sea posible hoy en día, y que hacen que las expectativas sean que cada día que pasa es más viable. Algunas de estos factores por separado también han ayudado al desarrollo de otros campos de investigación, pero la unión de todos ellos hacen que la computación ubicua pueda ser realidad [Mat02]:

4.2.1. La ley de Moore

En 1965 Gordon Moore afirmó que el número de transistores por pulgada en circuitos integrados se duplicaba cada año y que la tendencia continuaría durante las siguientes dos décadas.

Algo más tarde modificó su propia ley al afirmar que el ritmo bajaría, y la densidad de los datos se doblarían aproximadamente cada 18 meses. Esta progresión de crecimiento exponencial: doblar la capacidad de los microprocesadores cada año y medio, es lo que se considera la *Ley de Moore*.

Y esta ley se ha venido cumpliendo hasta el día de hoy, la capacidad de cómputo de los procesadores avanza muy rápidamente. Pero no solo la capacidad de cómputo de los procesadores, sino también la capacidad de almacenamiento, el ancho de banda para las comunicaciones, ... En resumen, cada poco tiempo tenemos dispositivos más baratos, más pequeños y más potentes. Y no parece que se vaya a parar este crecimiento, sino todo lo contrario, la previsión para los próximos tiempos es que siga ocurriendo lo mismo.

Pero hay un problema, y es que no todos los factores aumentan al ritmo de la *ley de Moore*, éste es el caso de la capacidad de almacenar energía mediante baterías. Esto supone un gran problema para estos dispositivos de los que hablamos, porque la capacidad de procesamiento, de almacenamiento,... crecen exponencialmente, y también, aunque no al mismo ritmo crece el consumo de energía, pero la capacidad para dotar a estos dispositivos de la energía necesaria crece muy lentamente. Este es un campo en el que todavía es necesario que se produzcan muchos avances.

4.2.2. Nuevos materiales

El desarrollo en el campo de los materiales también es muy importante. Hay muchos desarrollos en nuevos materiales que ya son estables y usados actualmente , pero también hay otro tipo de materiales que está actualmente en pleno desarrollo y que pueden presentar grandes avances para la computación ubicua:

- "Displays" flexibles: el uso de polímeros emisores de luz permite crear pantallas formadas por láminas de plástico muy finas, flexibles y plegables.
- Tinta electrónica y papel inteligente: que pretenden conseguir que el bolígrafo y el papel se conviertan en dispositivos verdaderamente móviles.
- El desarrollo de fibras informatizadas que se pueden entremezclar con los tejidos, con lo cual se pueden insertar transistores, sensores y unidades de procesamiento entre la estructura de la fibra.

Pero todavía queda algo de tiempo para que estas tecnologías puedan ser llevadas al campo de la práctica, lo cual supondrá un gran avance para el mundo de la computación ubicua.

4.2.3. Avances en la tecnología de la comunicación

Otro gran avance, como ya hemos comentado en los capítulos anteriores, es el avance en el sector de las comunicaciones:

- La fibra óptica ha aumentado la capacidad de las líneas de comunicaciones hasta poder establecer transmisiones de hasta Gigabits por segundo
- Tecnología de redes inalámbrica. También se han producido grandes avances en la telefonía móvil (GSM, UMTS) y en las redes locales inalámbricas (sección 2.3).
- Redes de área personal: ofrece la creación de pequeñas redes alrededor de los usuarios (sección 2.5).

4.2.4. Desarrollo de los sensores

El campo de los sensores también se ha desarrollado bastante en los últimos tiempos, tanto tecnológicamente como físicamente por el reducido tamaño que se ha conseguido en estos sensores. Algunos de estos avances son:

- Cámaras y micrófonos de muy reducido tamaño acompañado de reconocimientos de patrones y de técnicas de reconocimiento de voz
- Detectores de huellas digitales en objetos móviles
- Sensores de localización
- Dispositivos RFID: dispositivos para identificación por radiofrecuencia sin necesidad de contacto con el lector.

4.3. Escenarios

¿Cómo serán las situaciones en las que hipotéticamente se implantará la computación ubicua en un futuro? Pues no es fácil adivinar, pero para intentar comprender mejor en qué consiste este mundo de computación ubicua vamos a exponer algunas situaciones que se pueden dar ahora mismo o en un futuro, en las que se muestran las aportaciones de la computación pervasiva a la vida real, o a las posibles situaciones futuras. Algunas de estas situaciones puede que hoy en día ya sean reales.

Algunas de estas situaciones han sido simuladas dentro de las limitaciones que impone la tecnología actual, y de la disponibilidad de hardware con la que contábamos. Y el resto solamente son descritas planteando posibles situaciones futuras.

4.3.1. Seguimiento de personas

Tenemos la siguiente situación:

Situamos el escenario en una guardería que ofrece un servicio especial a los padres de los niños que entran en esta guardería. El servicio ofrecido, es que en todo momento los padres podrán ver a sus hijos mediante una página web, independientemente de donde se encuentren los pequeños.

Esta situación se puede dar hoy en día perfectamente, porque tenemos la tecnología necesaria para realizarlo. Una de las posibles soluciones a esta situación sería la siguiente:

En cada sala de la guardería tenemos un ordenador con una videocámara conectada, el ordenador está encendido 24 horas al día y la cámara está continuamente grabando y preparada para transmitir por videoconferencia cuando sea necesario.

Cuando un cliente quiere comenzar a recibir vídeo, se realiza una conexión entre el cliente y el servidor de vídeo, que es el ordenador de la sala en la que se encuentra el niño en ese momento. Cuando el niño cambia de habitación, hay que cambiar de servidor de la transmisión, para ésto utilizamos el protocolo de movilidad Mobile IP (sección 3.2), teniendo en cuenta que en todo momento el servidor que envía el vídeo es el nodo móvil, aunque en este caso lo que se mueve no es el ordenador, pero podemos hacer una aplicación para simular ese movimiento y hacer que la conexión entre el *correspondent node* (cliente de la transmisión de vídeo) y el nodo móvil sea continua en todo momento.

En realidad, en esta situación, lo que se mueve realmente es una persona, y necesitamos alguna forma de localizar en todo momento a esa persona (el niño) para poder utilizar el ordenador de la sala actual como nodo móvil de la comunicación mediante Mobile IP. Para localizar a esta persona se pueden utilizar varias técnicas, una de ellas puede ser que la persona lleve consigo un dispositivo RFID, que transmite señales inalámbricas por radiofrecuencia y tener receptores en las salas para recibir estas señales y saber en cada momento donde se encuentra el niño.

De este modo tenemos en todo momento localizado al niño y se puede utilizar el dispositivo de grabación de la sala en la que se encuentre para enviársela a los padres.

Esta misma situación puede aplicarse a muchos casos más, como por ejemplo para hacer el seguimiento de vehículos en una cadena de montaje, seguimiento

de presos en una prisión, ...

Esta situación la hemos simulado, con la tecnología con la que contamos y está desarrollada en la sección 4.4.1.

4.3.2. Información según la situación

Tenemos la siguiente situación:

Nos situamos en un museo en el que la dirección del museo ha decidido substituir los actuales guías que van proporcionando información sobre el contenido de cada sala o cada objeto, por dispositivos que llevarán los visitantes y en los que se les mostrará y podrán conseguir toda la información necesaria de cada sala sin tener que hacer nada más que ir visitando el museo. Cuando cambien de sala o se sitúen enfrente de un objeto se les mostrará y podrán escuchar la información pertinente.

Esta situación también es viable hoy en día con la tecnología con la que contamos.

Los dispositivos que proporciona el museo a los visitantes son PDAs con tecnología 802.11b (sección 2.4) para las comunicaciones inalámbricas y también receptor de infrarrojos (2.6). Todo el museo está cubierto con puntos de acceso para que desde cualquier sala haya conexión a la red de museo, o incluso a Internet.

En cada sala, y quizá en cada objeto, hay un dispositivo denominado *beacon*. Este dispositivo es un dispositivo muy pequeño que posee capacidad de comunicaciones inalámbricas, y que se utiliza usualmente para el envío de pequeñas cantidades de datos cada cierto periodo de tiempo. Normalmente la información enviada es una URL que se configura previamente en el dispositivo.

De esta forma en cada sala u objeto estos dispositivos envían una URL, que es captada por el receptor de infrarrojos del dispositivo móvil que llevan los visitantes, y a partir de esta URL, mediante HTTP y la red inalámbrica disponible con 802.11, los PDAs hacen una solicitud de esa información a un servidor web que está disponible en el museo y le muestran esta información a los visitantes, en forma de páginas HTML, vídeo o audio.

4.3.3. Continúa la videoconferencia

La situación sería la siguiente:

Un ejecutivo necesita tener videoconferencias muy a menudo con personas de otras ciudades, pero es una persona muy ocupada y necesita una movilidad continua dentro de la empresa en la que trabaja. Por eso necesita estar siempre disponible y poder realizar una videoconferencia en cualquier momento, y cambiar de sala o de ordenador mientras está teniendo estas videoconferencias.

La solución tecnológica podría ser la siguiente. El ejecutivo lleva siempre consigo un dispositivo PDA, con posibilidad de comunicaciones inalámbricas para conectarse a la red de la empresa y con software para realizar videoconferencias. Este dispositivo será el nodo móvil para utilizar la tecnología de movilidad Mobile IP, por tanto aunque se esté moviendo podrá seguir teniendo la videoconferencia.

Pero también necesita que cuando llegue a su despacho, o sala similar, la videoconferencia pase a tener lugar desde el ordenador de sobremesa del que dispone la sala. Para ello esos ordenadores de sobremesa cuentan con dispositivos

dongles (sección 2.6.3), que detectan que el PDA está al lado de ellos mediante el puerto de infrarrojos, y se simula el cambio del nodo móvil a ese ordenador de sobremesa para poder seguir con la videoconferencia desde él.

De esta forma, puede utilizar ordenadores de sobremesa cuando tenga disponible alguno, y si esto no ocurre utilizará su dispositivo móvil.

Esta situación se puede modelar con la información disponible en el apéndice 4.4.2.

4.3.4. Charla en sala "pervasiva"

Situación:

Un profesor en una universidad ha improvisado una charla para unos estudiantes. El profesor lleva un PDA en el que lleva toda la información para la charla, la presentación, unos apuntes, ... y necesita compartir esta información con los alumnos, necesita proyectar las diapositivas con el proyector disponible en la sala, ...

Para esta situación hay que hacer uso de las redes ad-hoc (sección 2.8). Gracias a este tipo de tecnología, se puede crear una red local al momento entre el dispositivo inalámbrico del profesor, los que dispongan los alumnos y los elementos "ubicuos" que existan en la sala se puedan comunicar sin problemas. Esta red ad-hoc se puede crear entre dispositivos con tecnología 802.11, u otros tipos de tecnología como Irda (sección 2.6).

Por tanto, dependiendo de los dispositivos que tengan cada uno de ellos utilizarán uno de estos tipos de comunicación o incluso ambos. Si tienen portátiles con tarjetas 802.11b crearán la red local con ellas, que sería la opción recomendada porque sería más fácil crear cobertura para todos ellos en esta red, o sino pues también podrían usar los dispositivos infrarrojos de los portátiles o PDAs.

La sala contaría con otros dispositivos "ubicuos", como una pizarra electrónica o un proyector en la que se podrían mostrar las diapositivas del profesor, u otro tipo de información, porque estos dispositivos "ubicuos" también formarían parte de la red ad-hoc creada en la sala que estuvieran para facilitar las comunicaciones entre todos ellos.

4.4. Prácticas sobre Computación Ubicua

A continuación se definirán algunas de las simulaciones realizadas de posibles escenarios de computación ubicua. Al ser simulaciones, algunos de los dispositivos utilizados no se corresponden con los que se usarían en la aplicación real de estos escenarios, pero estos dispositivos son similares en cuanto a funcionalidad.

4.4.1. Seguimiento de personas

Planteamiento

La simulación consiste en lo siguiente: Hay una máquina que está recibiendo una videoconferencia, y hay otras máquinas transmiten este tráfico. Estas máquinas se supone que realizan un seguimiento de un objeto o persona y se intercambian el papel de emisor dependiendo de donde se encuentre el supuesto objetivo.

Para ello se utiliza la implementación de Mobile IPv4 instalada de la forma explicada en la sección 3.5.1, pero sobre la maqueta de la figura 4.1

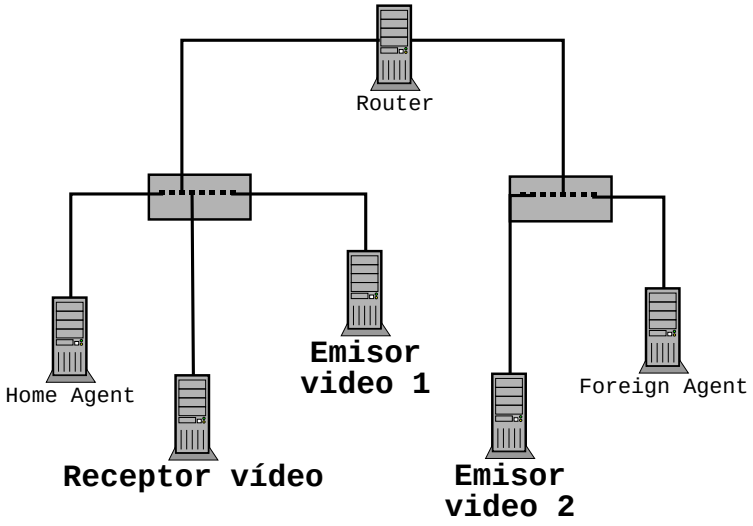


Figura 4.1: Maqueta para la simulación de seguimiento

Como podemos ver en la figura 4.1, tenemos un ordenador que recibe el vídeo, en este caso se encuentra en la *home network* de Mobile IP, pero podría estar en cualquier otra red. En esta misma red tenemos al *home agent*, y a uno de los nodos que actuarán como receptores de la transmisión. En la otra red (*foreign network* de Mobile IP) tenemos un *foreign agent* y al otro nodo encargado de la transmisión de vídeo.

Cada uno de los emisores de vídeo poseen una videocámara grabando continuamente, y tanto los emisores como el receptor tienen un software para realización de videoconferencias llamado vic.

Procedimiento

El funcionamiento sería el siguiente: Inicialmente hay una videoconferencia entre el receptor y el emisor 2. Este emisor 2 posee la dirección IP del nodo móvil que actúa en el protocolo Mobile IP. Cuando queremos cambiar el emisor deberíamos dejar de transmitir por el emisor 2 y empezar a recibir vídeo del emisor 1.

Para ello se hace una simulación de lo que sería el movimiento de un nodo móvil de una red a otra en Mobile IP, pero en esta ocasión no movemos el nodo, sino que intercambiamos las direcciones IP entre las máquinas emisoras para que el protocolo Mobile IP crea que el nodo se ha cambiado de red.

De esta forma cuando queramos recibir de uno de los emisores necesitamos que le sea asignada la dirección IP del nodo móvil, y de esta forma continúe la transmisión de vídeo hacia el receptor. Para realizar esta simulación de movimiento de nodo móvil en el que se intercambian las direcciones IP, hemos realizado unos programas en el lenguaje de programación Ada usando la librería de comunicaciones Lower Layer.

Implementación

La arquitectura de los programas sigue el modelo cliente/servidor:

- Servidor: El servidor es lanzado en la máquina que actúa como receptora de vídeo. La función del servidor es esperar hasta recibir un mensaje de alguno de los emisores de vídeo, cuando recibe el mensaje de uno de ellos (lo envía porque quiere transmitir) envía un mensaje al emisor que actualmente está emitiendo para que intercambie las direcciones IP con el que nodo que lo ha solicitado. Después de esto sigue esperando a que otro nodo solicite transmitir.
- Cliente: El cliente se lanza en cada uno de las máquinas que están listas para transmitir, en nuestro caso los dos emisores. Este programa espera hasta que recibe una señal, esta señal podría ser la recepción de algún mensaje por infrarrojos o algún tipo de sensor. En nuestro caso la señal que recibe el cliente es la pulsación de la tecla "Enter".

  Cuando un cliente recibe esta señal envía un mensaje al servidor para que le indique al nodo que transmite actualmente que deje de hacerlo y que intercambie las direcciones IP con él. De esta forma el emisor que estaba transmitiendo deja de hacerlo y el nuevo emisor sigue transmitiendo al servidor.

Hay que tener en cuenta que el uso de Mobile IP es imprescindible, porque el receptor de vídeo en todo momento está teniendo una videoconferencia supuestamente con una sola máquina en todo momento, pero gracias a este protocolo podemos cambiar la dirección a otra máquina en otra red y que la videoconferencia siga en curso.

Simulación

Pasos a seguir para realizar la simulación:

- En el *home agent* es necesario arrancar el software de Mobile IP:

  ```
  # dynhad --fg --debug
  ```

- En el *foreign agent* es necesario arrancar también el software de Mobile IP:

  ```
  # dynhad --fg --debug
  ```

- En el receptor de vídeo hay que arrancar el software de la aplicación servidor y el software para recibir el vídeo (indicando la dirección IP del nodo móvil de Mobile IP):

  ```
  # ./servidor
  # vic 192.168.242.2/8888
  ```

- En el emisor que comienza retransmitiendo vídeo hay que arrancar el software de Mobile IP, el software de videoconferencia (con la dirección IP del servidor) y el software cliente de la aplicación:

```
# dynmnd --fg --debug
# vic 192.168.242.3/8888
# ./cliente -mn
```

- En los demás emisores hay que arrancar el software cliente de la aplicación:

```
# ./cliente
```

Una vez realizados estos pasos el servidor estará recibiendo del emisor que hayamos configurado. Para cambiar de emisor bastará con pulsar la tecla "Enter" del ordenador que queremos que siga transmitiendo.

4.4.2. Movilidad de personas

Planteamiento

En este caso, la simulación consiste en lo siguiente: hay una máquina en una determinada red que está transmitiendo vídeo, y hay otra máquina o dispositivo que quiere recibir esa transmisión mientras se mueve por varias redes diferentes.

Para la simulación de este escenario se ha utilizado la implementación de Mobile IPv6 configurada tal cual se muestra en la sección 3.7.1, y se ha utilizado la misma maqueta, podemos verla de nuevo en la figura 4.2.

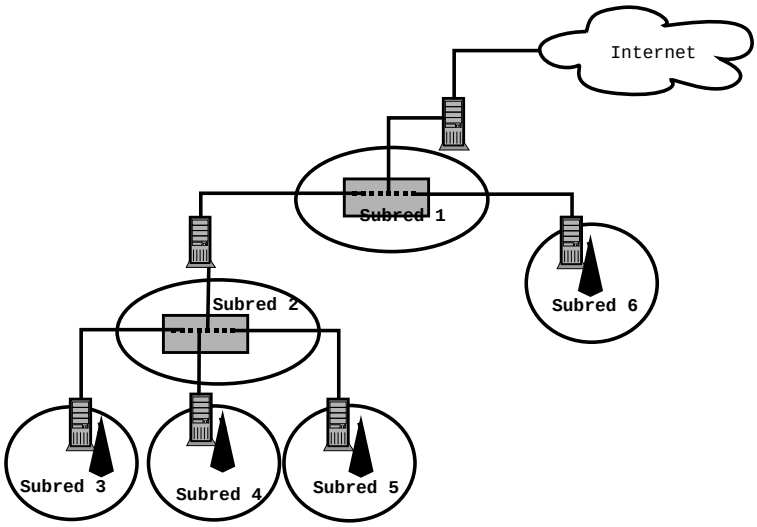


Figura 4.2: Maqueta para la simulación de movilidad

Habrá un nodo emisor de vídeo en una de las subredes de la maqueta o en otra red de fuera de la maqueta, ya que esta maqueta está conectada a Internet mediante el gateway de la subred 1.

Y habrá un nodo receptor que es un ordenador portátil, que estará en alguna de las subredes de la maqueta que recibirá la transmisión de vídeo. Estos dos nodos disponen del software para videoconferencia llamado vic.

Procedimiento

El procedimiento sería el siguiente: Inicialmente el nodo emisor (*correspondent node* de Mobile IPv6) está transmitiendo vídeo hacia el nodo receptor. Este nodo receptor actúa como nodo móvil de Mobile IPv6, y estará por alguna de las *foreign networks* 3, 4, 5 o 6, que son las que disponen de *foreign agents* para dar conexión a los nodos móviles. El *home agent* está situado en alguna red de fuera de la maqueta, aunque también podría estar en alguna de las de la maqueta.

Cuando el nodo móvil se mueva de una subred a otra, informará al *home agent* de su nueva situación y de esta forma podrá seguir recibiendo el la señal de vídeo independientemente de la red a la que se conecte.

Simulación

Para la simulación de este escenario es necesario seguir los pasos que se describen en la sección 3.7.2 para iniciar el funcionamiento de Mobile IPv6 en la maqueta, y arrancar el software de videoconferencia en el cliente y en el servidor de la transmisión:

- Cliente:

```
# vic fec0::3:2e0:4cff:fe69:2d78/8888
```

- Servidor:

```
# vic fec0::7:260:1dff:fef1:2be9/8888
```

Después de esto, solamente con ir moviendo el ordenador portátil que actúa como nodo móvil entre las diferentes subredes podremos ver que podemos seguir con la videoconferencia independientemente de donde estemos.

Capítulo 5

Conclusiones

La computación ubicua, como hemos visto a lo largo de todo el estudio realizado, ofrece una visión de futuro con una gran cantidad de posibilidades en la vida cotidiana de las personas. La idea de computación ubicua se extiende sobre todos los dispositivos u objetos que nos rodean y plantea nuevas formas de entender nuestro entorno, y nuevos modos de interactuar con todos estos objetos, tanto nosotros con ellos como ellos con nosotros.

Como hemos visto, la computación ubicua todavía es un campo muy joven, un campo por descubrir en el que todavía queda mucho por investigar, pero no solo en el campo de la computación ubicua en sí, sino en todos los campos en los que se basa: tecnología de componentes, sensores, comunicaciones, nuevos materiales, y muchos más. Por tanto, aunque ciertos aspectos de la computación pervasiva ya pueden ser llevados a la práctica, el futuro de esta tecnología depende del desarrollo de todos estos campos mencionados, según vayan avanzando estos campos irán abriendo nuevas posibilidades de expansión a la computación ubicua.

Pero el objetivo final, no es solamente que esta tecnología se desarrolle y se introduzca en todos los objetos cotidianos. El objetivo final es que la computación ubicua pase desapercibida, es decir, que la gente que usa estos dispositivos "ubicuos" no sepa realmente toda la tecnología que se encuentra detrás de los dispositivos u objetos que usan, que los usuarios se centren solamente en las acciones que deben realizar y no en las herramientas que usan para ello. En resumen, que la computación ubicua pase a un segundo plano dejando en el primer plano a los usuarios y a las tareas que realizan, no a los dispositivos que utilizan.

Hay muchos factores que influyen en el futuro de este campo, y no solo son factores tecnológicos, hay otro tipo de factores que influyen más que estos últimos, son los factores económicos y sociales. Es imposible predecir si todos estos avances serán viables económicamente, y si serán socialmente bien aceptados. Hay que tener en cuenta que la computación ubicua introduce muchos avances que pueden influir en la privacidad y en la intimidad de las personas si no se hace un buen uso de ellos.

Por tanto, el factor social y económico, e incluso político debe ser tenido en cuenta a la hora de valorar el futuro de esta tecnología, porque ya se han dado hechos en la historia de tecnología superior a otra existente (sistemas Beta vs VHF, por ejemplo) que no tuvieron éxito por razones no tecnológicas.

En cualquier caso, tanto si se extiende en un futuro como si no lo hace, lo que está claro es que la computación ubicua o pervasiva ofrece un cambio de percepción de las tareas realizadas diariamente que proporcionaría grandes avances para los usuarios finales, que sería todo el mundo, facilitándole todas estas tareas y ofreciendo nuevas posibilidades hoy difíciles de predecir.

Al finalizar el proyecto, basándonos en los objetivos planteados en la sección 1.2, puedo afirmar que se ha conseguido la realización de todos estos objetivos. Se ha realizado un estudio teórico de toda la tecnología propuesta, se han realizado las pruebas prácticas sobre toda esta tecnología sin demasiadas dificultades y se ha experimentado con una amplia variedad de diversos dispositivos. Específicamente, podríamos destacar las siguientes conclusiones:

- La tecnología inalámbrica está muy avanzada, sobre todo el estándar 802.11b, y se puede introducir en la mayoría de los dispositivos móviles de los que disponemos en la actualidad de forma sencilla, quizá por esto parece que a día de hoy es el estándar que está siendo socialmente más aceptado, y por ello en muchas empresas se crean redes con esta tecnología, y también se están desarrollando mucho las redes inalámbricas ciudadanas con tecnología 802.11b.

- Los PDA también están siendo desarrollados muy rápidamente, y proporcionan muchas posibilidades para a los usuarios. Pero claramente, la ventaja entre los dos tipos de PDA que hemos probado durante los experimentos, ha sido que uno de ellos (Compaq Ipaq) nos permitía cambiar el sistema operativo, lo que le añadía muchas más posibilidades de uso, un aumento del rendimiento y de la funcionalidad del mismo. Además estos dispositivos vienen cada día mejor equipados, sobre todo en el aspecto de las comunicaciones, proporcionan posibilidad de comunicación por el puerto de infrarrojos, puerto serie, puerto usb e incluso la inclusión de dispositivos para comuncaciones por 802.11 o Bluetooth.

- Las implementaciones de Mobile IPv4 y Mobile IPv6 que hemos utilizado en las maquetas están bastante desarrolladas y su estado de desarrollo es muy estable, además el desarrollo de las mismas sigue muy activo. La implantación de la movilidad usando estas implementaciones es relativamente sencillo y su rendimiento es bastante aceptable, pero en la vida real parece que no es muy usado. Así como las redes inalámbricas 802.11b son muy usadas y su uso sigue creciendo, el uso de la tecnología de movilidad no es usado prácticamente en ambientes que sean de investigación.

- Las implementaciones de Cellular IPv4 y Cellular IPv6 no son tan estables como las de Mobile IP, y su desarrollo no está tan avanzado, quizá porque el uso de estas implementaciones está menos extendido, incluso en el ámbito de la investigación. Durante la implantanción que nosotros hicimos de esta tecnología en las maquetas montadas tuvimos algunos problemas, que sin ser problemas graves que pudimos arreglar, demostraban que el desarrollo de estas implementaciones no está tan avanzado.

- Cellular IP y Mobile IP deben ser implementados como definen los estándares para proporcionar compatibilidad entre ellos para poder interactuar, pero las pruebas realizadas en este aspecto no fueron nada satisfactorias

porque las implementaciones probadas entre Cellular IP y Mobile IP no interactuaban entre ellas, y hubo que hacer una gran cantidad de cambios para que pudieran interactuar mínimamente, sin conseguir los resultados deseados con el uso de ambas tecnologías.

- La implementación del protocolo DSR para la arquitectura de los robots Legos se desarrolló bajo la arquitectura i386, por tanto hubo que utilizar compilación cruzada para su desarrollo, pero esta forma de desarrollo es muy habitual y está bien preparada, por lo que no hubo mayores dificultades al realizar este desarrollo. Algunos inconvenientes fueron que, al tener que desarrollar el protocolo para la arquitectura legos y para la arquitectura i386, algunas partes de la implementación eran dependientes de la arquitectura (como el tema de threads o las librerías de comunicaciones inalámbricas) y se tuvieron que realizar estas dos implementaciones casi en paralelo para poder funcionar sobre las dos arquitecturas.

- La experimentación sobre las redes ad-hoc desarrolladas con los robots Legos fue muy interesante, y nos mostró el funcionamiento real de este tipo de redes, y también pudimos observar algunos de los problemas con los que nos encontramos, como las colisiones en las comunicaciones por infrarrojos o el aumento del consumo de energía cuando las comunicaciones inalámbricas aumentan.

Después de todo esto se puede afirmar que el nivel de desarrollo de toda esta tecnología está muy avanzado, ya que hemos conseguido realizar experimentos sobre los temas que hemos elegido sin grandes problemas, aunque todavía queda mucha investigación y desarrollo por realizar para mejorarlos.

Personalmente, con el desarrollo del proyecto he conseguido varias cosas. Una de ellas ha sido consolidar los conocimientos adquiridos durante los años de estudio en la universidad, sobre todo en el campo de las redes y las comunicaciones, pero en general me ha servido para aprender a estudiar sobre temas nuevos en base a los conocimientos adquiridos durante los estudios en las asignaturas. Otro logro interesante ha sido el poder realizar estudios sobre tecnología que se encuentra en plena investigación en la actualidad, y el poder haber realizado experimentos con dispositivos que todavía no son muy frecuentes porque forman parte de esas investigaciones actuales. Es decir, me ha permitido realizar una labor de investigación sobre campos de actualidad.

5.1. Desarrollo del proyecto

El proyecto se ha realizado durante aproximadamente 2 años, aunque durante estos dos años ha habido periodos de más actividad y otros periodos con actividad escasa.

El tiempo real estimado para la realización del proyecto es de unas 600 horas, aproximadamente repartidas entre las siguientes tareas:

- Estudio teórico de la tecnología inalámbrica (5 % del total)

- Configuración y prácticas de dispositivos inalámbricos: Infrarrojos y 802.11 de los Compaq Ipaq, infrarrojos en el HP Jornada, dongles, beacons, ... (10 % del total)

- Estudio teórico de los protocolos de movilidad (5 % del total)
- Montaje de maqueta de pruebas para Mobile IPv4 (15 % del total)
- Instalación y pruebas con Cellular IPv4 (5 % del total)
- Montaje de maqueta de pruebas para Mobile IPv6 y pruebas de rendimiento (20 % del total)
- Instalación y pruebas con Cellular IPv6 (5 % del total)
- Estudio teórico de la computación ubicua (5 % del total)
- Implementación del protocolo DSR para la arquitectura de robots Legos (10 % del total).
- Simulaciones de escenarios del modelo de computación ubicua (10 % del total)
- Generación de la documentación del proyecto (10 % del total)

Algunas de las labores realizadas durante el proyecto se han realizado conjuntamente con otras personas. Por ejemplo, el montaje de maquetas de redes para pruebas de protocolos de movilidad se ha realizado junto con Raúl Rodríguez Aparicio. Y la implementación del protocolo DSR para los robots Legos se ha realizado junto con: José Pelegrín y Raúl Rodríguez.

Apéndice A

Glosario

- **Beacon**: Dispositivo de comunicaciones inalámbricas por infrarrojos con una capacidad de enviar datos con ancho de banda muy reducido. Usualmente utilizado para emitir una URL.
- **Computación ubicua**: Campo de la computación que se basa en la idea de introducir la capacidad de computación y comunicación en todos los objetos cotidianos, e intentar que el mundo de la computación pase a un segundo plano, para que los usuarios se centren en las tareas a realizar y no en las herramientas utilizadas.
- **Dongle**: Dispositivo para comunicaciones inalámbricas. Normalmente se conecta a un ordenador mediante el puerto serie o USB para dotar al ordenador de posibilidades de comunicaciones inalámbricas mediante IrDA (sección 2.6).
- **ISM**: En inglés: *Industrial, Scientific, Medical*. Se corresponde con un rango de frecuencia del espectro de comunicaciones por radio que está liberado para uso libre sin necesidad de licencias.
- **LAN**: En inglés: *Local Area Network*: Es una red de ordenadores o dispositivos computaciones dentro de un área reducida.
- **LLC**: En inglés: *Logical Link Control*: Es una de las dos subcapas que forman la capa de control de acceso de datos en la pila de protocolos de IEEE 802. Se encarga de: gestionar el enlace de datos en la comunicación, definir los servicios de los punto de acceso, ...
- **MAC**: En inglés: *Media Access Control* Es una de las dos subcapas que forman la capa de control de acceso de datos en la pila de protocolos OSI. Se encarga de mover los datos desde una tarjeta de interfaz de red a otro.
- **MAN**: En inglés: *Metropolitan Area Network*: Es una red de datos diseñada para una ciudad. En términos de tamaño es más grande que una LAN, pero más pequeña que una WAN. Está diseñada para disponer de conexiones de alta velocidad.
- **PAN**: En inglés: *Personal Area Network*: Es una tecnología diseñada para que una persona se pueda comunicar de forma inalámbrica con los dispositivos que le rodean: teléfono, PDA, cascos para oir música, ...

- **PDA**: En inglés: *Personal Digital Assistant*. Es un dispositivo de mano, de reducidas dimensiones, que puede incorporar: teléfono, conexión a internet, comunicaciones inalámbricas, ... Además de poseer software de características limitadas, aunque cada día están más desarrollados.
- **RFID**: En inglés: *Radio Frecuency Identification*: Pequeño dispositivo que emite señales de radiofrecuencia. Suele ser usado para los mismos fines que los códigos de barras, pero tiene la ventaja de que no es necesaria una visión directa del lector, y puede realizarse a mayores distancias.
- **WAN**: En inglés: *Wide Area Network*: Consiste en una red localizada en un área geográfica muy extensa, normalmente está formada por unas cuantas LAN.
- **WEP**: En inglés *Wired Equivalent Privacy*: Protocolo de seguridad para WLAN definido en el estándar 802.11b. Se basa en el cifrado de datos que son enviados por la red. Se ha comprobado que no es tan seguro como se diseñó.
- **WLAN**: En inglés *Wireless Local Area Network*: Es un tipo de LAN que utiliza comunicaciones inalámbricas en lugar de cableado.

Bibliografía

[Ara95] Agustin A. Araya. Questioning ubiquitous computing. In *Proceedings of the 1995 ACM 23rd annual conference on Computer science*, pages 230–237. ACM Press, 1995.

[Ass93] Infrarred Data Association. Irda, 1993. http://www.irda.org.

[Bel58] R. Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16(1):87–90, 1958.

[CMU] Carnegie Mellon University CMU. Project aura. http://www-2.cs.cmu.edu/ aura/.

[Fam01] Familiar distribution, 2001. http://familiar.handhelds.org/.

[Fun02] Free Software Fundation. Gfdl license, November 2002. http://www.gnu.org/licenses/fdl.txt.

[HP] Hewlett Packard HP. Cooltown. http://www.cooltown.hp.com.

[HUT99] Helsinki University Of Technology HUT. Dynamics - hut mobile ip, 1999. http://www.cs.hut.fi/Research/Dynamics/.

[Mat02] Friedemann Mattern. Ubiquitous & pervasive computing: A technology-driven motivation. 8 2002. http://www.inf.ethz.ch/vs/publ/slides/dag2002-mattern-1.pdf.

[MIT] Massachusetts Institute Technology MIT. Oxygen. http://oxygen.lcs.mit.edu.

[PC95] T.; Pahlavan, K.; Probert and M. Chase. Trends in local wireless networks. *IEEE Communications Magazine*, Marzo 1995.

[Per96] Charles E. Perkins. Ip mobility support, 1996.

[Per97] Charles E. Perkins. *Mobile IP; Design Principles and Practices.* Addison-Wesley Longman Publishing Co., Inc., 1997.

[Per01] Charles E. Perkins. *Ad-hoc networking.* Addison-Wesley Longman Publishing Co., Inc., 2001.

[RT99] E. Royer and C. Toh. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications Magazine*, 6(2):46–66, 4 1999.

[Sat95] M. Satyanarayanan. Pervasive computing: Vision and challenges. *IEEE Personal Communications*, 8(4), 8 1995.

[SK01] Javier Gomez Sanghyo Kim. Cellular ipv6, 2001. http://w3dpdext1.intranet.gr/cipv6/download_area.htm.

[Sol] James D. Solomon. *Mobile IP: The Internet Unplugged*. Prentice Hall.

[Sta01] William Stallings. *Wireless Communications and Networks*. Prentice Hall Professional Technical Reference, 2001.

[TL00] HUT Telecommunications and Multimedia Lab. Mipl mobile ipv6 for linux, 2000. http://www.mipl.mediapoli.com/.

[Toh02] C.-K. Toh. *Ad hoc mobile wireless networks: protocols and systems*. Prentice Hall, 2002.

[Uni99] Comet Group In Columbia University. Cellular ip for linux, 1999. http://comet.ctr.columbia.edu/cellularip/.

[Wei91] Mark Weiser. The computer for the 21st century. *Sci. Amer.*, Septiembre 1991.

[Zim80] H. Zimmerman. Osi reference model the iso model of architecture for open systems interconnection. *IEEE Transactions on Communications*, 28(4):425–432, Abril 1980.