Complu6IX

Transición a IPv6 de la Red de Datos del Campus de Moncloa

Ingeniería de Red

Alberto Abián Belmonte

Jaime Frutos Morales

Javier Santamaría Reinoso

Complu6IX: Transición a IPv6 de la Red de Datos del Campus de Moncloa; Ingeniería de Red
by Alberto Abián Belmonte, Jaime Frutos Morales, and Javier Santamaría Reinoso

Table of Contents

List of Tables

Abstract

Este proyecto comprueba el estado de madurez del protocolo de red IPv6 y su viabilidad a la hora de realizar una migración de la red de la Complutense a dicho protocolo. Utiliza virtualización en GNU/Linux mediante VNUML para utilizar programas reales sobre una plataforma GNU/Linux simulada.

Prólogo

1. Motivación

Hace 10 años, IPv6 era considerado un experimento con el que se había practicado de modo simulado utilizando IPv4 como medio de transporte. Hoy en día, la conectividad IPv6 nativa está disponible a lo largo de diversos puntos en Internet, incluyendo áreas como Norte América, Europa, y Asia. En Europa, la red GEANT ha ofrecido una red de IPv6 dual stack desde 2003 y la mayoría de Redes Nacionales para la Educación e Investigación han desplegado IPv6 como un servicio disponible más. Estos esfuerzos han sido impulsados por los resultados producidos por 6NET y Euro6IX, entre otros. En España, RedIris es el encargado de suministrar a las Universidades el acceso a Internet.

El equipo Complu6IX ya ha experimentado con la conectividad IPv6 en el seno de la Red de Datos Complutense, practicando túneles que utilizan IPv4 como medio para transmitir datos en IPv6. En Internet, son varias las instituciones que prestan servicios como proveedores de túneles a IPv6 (Sixxs, Hurricane, Freenet6, etc).

Como es lógico, en un escenario provisto de conectividad IPv6 se eliminan gran parte de los problemas originados por conexiones mediante túneles, al ser éstos soluciones provisionales mientras se realiza una migración completa hacia una red IPv6 nativa. Por otra parte, el rendimiento de IPv6 es superior al de IPv4. Aunque en el mundo comercial es todavía un objetivo a largo plazo, el reto consiste en desplegar IPv6 en las universidades y, desde ahí, demostrar las nuevas ventajas de los servicios y aplicaciones sobre IPv6.

En nuestra opinión, la transición a IPv6 es inevitable y, en consecuencia, se recomienda un despliegue progresivo tomando las medidas necesarias desde el principio. Toda universidad debería desplegar servicios de red IPv6 como apoyo a la docencia y la investigación, al tiempo que supone un seminario ideal para la innovación, donde las tecnologías se presentan a la comunidad universitaria en general.

2. Agradecimientos

Nos gustaría mostrar nuestro agradecimiento a los siguientes grupos y centros:

- Centro de Proceso de Datos de la UCM: Por su apoyo desde el principio y su aprobación final cuando les presentamos el proyecto acabado.
- Departamento de Ingeniería de Sistemas Telemáticos: Por programar el simulador VNUML (Virtual Network User Mode Linux), la base de todo nuestro proyecto.
- Comunidad de programadores del Kernel Linux: Por dotar al kernel Linux de la virtualización mediante UML (User Mode Linux) sobre la que está basada VNUML.

Chapter 1. Introducción

1.1. Historia de las redes de computadores e Internet

En 1957, el Departamento de Defensa de Estados Unidos funda la organización ARPA (Agencia de Proyectos de Investigación Avanzados) para intentar mantener el liderazgo tecnológico frente a la Unión Soviética. En 1969, la agencia ARPA desarrolla la primera red de computadores, denominada ARPANET; una red de conmutación de paquetes basada en nodos de conmutación denominados IMPs (Interface Message Processors). La red ARPANET original precursora de la actual Internet, interconectaba cuatro centros de investigación: Universidad de California en los Ángeles, el Instituto de Investigación de Stanford, la Universidad de California en Santa Bárbara y la Universidad de Utah.

Durante la década de los 70 proliferan un gran número de redes comerciales basadas en la técnica de conmutación de paquetes: SNA de IBM (1974), DECNet de DIGITAL (1975), DCA de Univac (1976), TRANSDATA de Siemens (1978), DSA de Bull (1979), etc. Dada la diversidad de redes, todas ellas incompatibles entre si, la organización ISO (Internacional Standards Organization) publica en 1983 el modelo OSI (Open System Interconnection) o modelo de interconexión de sistemas abiertos como un marco de referencia para la definición de estándares compatibles para redes de conmutación de paquetes.

En la década de los 70 también se pone en funcionamiento la primera red de difusión o de canal compartido: la red ALOHA. Se trata de una red de conmutación de paquetes por radiofrecuencia desarrollada en la universidad de Hawai. Esta primera red de difusión puede considerarse la precursora de las actuales redes de área local (LAN, Local Área Network).

Basándose en muchas de las ideas aplicadas a la red ALOHA, la compañía Xerox desarrolla, a finales de los 70, la red Ethernet, la primera red de área local con topología de bus lineal sobre un cable coaxial. A principios de los 80 aparecen otros tipos de redes de área local: Token Ring de IBM, Token Bus de General Motors, Apple Talk de Apple, etc. En 1985, el comité 802 de la organización IEEE publica un conjunto de estándares para redes de área local que incluyen entre otros la definición de la norma 802.3 para redes de tipo Ethernet, la norma 802.4 para redes de tipo Token Bus y la norma 802.5 para redes de tipo Token Ring.

En 1983, ARPANET sustituye los protocolos nativos usados en los IMPs por la arquitectura de protocolos TCP/IP que, al ser independientes de la tecnología de la red, hacen posible que redes diferentes se puedan interconectar entre si. Los antiguos IMPs evolucionan hasta convertirse en los actuales routers o encaminadores y se acuña el termino Internet.

Durante la segunda mitad de la década de los 80 se desarrollan nuevas tecnologías de comunicación basadas en fibra óptica que dan lugar a las primeras redes de área metropolitana (MAN, Metropolitan Area Networks) de alta velocidad , tales como FDDI capaces de transmitir a 100 Mbps. En esta época surge también la idea de la red digital de servicios integrados (RDSI o ISDN, Integrated Services Digital Network), como una red capaz de integrar el sistema telefónico, la transmisión de datos, la transmisión

de audio y video, etc.

A principios de los 90, con la aparición de la tecnología ATM (Asynchronous Transfer Mode o modo de trasferencia asíncrono), surgen las redes RDSI de banda ancha (también denominadas B-ISDN o Broadband ISDN) basadas en ATM, que ofrecen mayores prestaciones que las RDSI originales.En paralelo al desarrollo de todas estas tecnologías la red Internet se populariza enormemente y se extiende por todo el mundo.

Durante la década de los 80 y 90, Internet experimenta un crecimiento imparable pasando de 1000 computadores conectados en el año 1984 a 10.000 en 1987, 500.000 en 1991, 5.000.000 en 1995, 30.000.000 en 1998 y cerca de 100.000.000 en el año 2000. En particular, en España la primera conexión a Internet se realiza en 1990 a través del servicio experimental de RedIRIS y en 1991 se conectan los principales centros universitarios (UCM, UPM, UAM, UPC, UAB) y de investigación (CIEMAT, CNM, CSIC, FUNDESCO, RICA). En este año se registran en España 1000 computadores conectados a Internet, pasando a 20.000 en 1994, 100.000 en 1996 y mas de 300.000 a finales de 1998.

A lo largo de la década de los 90, las redes de área local de tipo Ethernet experimentan un importante auge, gracias a su bajo coste, la sencillez de su instalación y sus buenas prestaciones (10 Mbps en su implementación original). En esta década surgen variantes de Ethernet de alta velocidad (Fast Ethernet a 100 Mbps y Gigabit Ethernet a 1000 Mbps) y estas tecnologías pasan a convertirse en las mas ampliamente extendidas en el marco de redes locales, en detrimento de otros tipos de redes (Token Ring y especialmente Token Bus), tendencia que continua en la actualidad.

1.2. Introducción a las redes de computadoras

1.2.1. Tipos de redes según su interconexión

1.2.1.1. Redes de enlace punto a punto dedicados

En este tipo de redes, cada pareja de ordenadores se comunica mediante un cable dedicado que interconecta ambos equipos. El caso mas sencillo que se puede plantear son dos ordenadores personales directamente conectados a través de sus respectivos puertos serie o paralelo, mediante un cable de conexión debidamente configurado (las líneas de transmisión y recepción deberán estar cruzadas) y un software especifico de comunicación. Si se desea una mayor velocidad de transmisión, es posible instalar una interfaz de red que admita conexión directa (por ejemplo Fast Ethernet).

Este tipo de conexión puede ser útil para conectar un numero reducido de equipos (dos o a lo sumo tres), cuando el numero de equipos crece, esta solución se vuelve inviable por dos razones fundamentales. La primera, es que en cada equipo son necesarios un gran número de puertos de comunicación y la segunda es que se necesita un cable de conexión dedicado por cada pareja de ordenadores, por lo que el número de enlaces crece rápidamente con el número de equipos. En particular si el número de equipos de la red es N, el número de enlaces necesarios será igual a N*(N-1)/2.

1.2.1.2. Redes de conmutación de circuitos

La conexión mediante enlaces punto a punto dedicados también se empleo en las primeras líneas telefónicas. No obstante, ante el rápido crecimiento del número de usuarios y la inviabilidad de mantener este tipo de conectividad, surgieron rápidamente las centralitas telefónicas, dando lugar a las redes telefónicas conmutadas o redes de conmutación de circuitos, esquema que se sigue utilizando en la actualidad.

Las redes de conmutación de circuitos están formadas por una serie de nodos de conmutación de circuitos, también denominadas centralitas de conmutación, que establecen las conexiones necesarias para hallar un camino físico entre los extremos que se quieren comunicar.

Aunque la conmutación de circuitos habitualmente se emplea en telefonía para la transmisión de voz, también puede utilizarse para la transmisión de datos usando una interfaz de transmisión recepción adecuada en cada computador (en general, un módem) que transforma las señales de datos digitales generadas por el computador en señales admisibles por la línea de comunicación.

En el proceso de comunicación a través de una red de conmutación de circuitos se distinguen tres etapas: Fase de establecimiento de la conexión o llamada, fase de transmisión y fase de terminación de la conexión.

Las redes de conmutación de circuitos presentan algunas desventajas como que la conexión se paga, habitualmente, durante todo el tiempo que se mantiene establecida, así como que las conexiones en este tipo de redes son dedicadas, es decir, cuando se establece un circuito entre dos equipos, este no puede compartirse con otros usuarios, incluso en el caso de que la conexión establecida no se utilice durante determinado periodo de tiempo.

1.2.1.3. Redes de conmutación de paquetes

Las redes de conmutación de paquetes están especialmente diseñadas para la transmisión de datos y resuelven los inconvenientes de las redes de conmutación de circuitos. En este tipo de redes la información a enviar se divide en pequeños bloques de un tamaño limitado llamados paquetes o datagramas. Cada equipo conectado a la red se identifica mediante una dirección única y normalmente cada paquete enviado debe llevar la dirección del equipo emisor y el equipo destinatario.

Estas redes están formadas por un conjunto de nodos de conmutación de paquetes (PSNs Packet Switching Nodes) interconectados entre si, a los cuales, a su vez, se conectan los computadores de la red. Cuando un PSN recibe un paquete, examina la dirección de la maquina destinataria y, en función de esta, lo reenvía al PSN mas adecuado, de manera que el paquete va pasando de un PSN a otro, hasta alcanzar su destino. Esta función recibe el nombre de encaminamiento o routing.

En este tipo de redes es posible facturar por volumen de datos transmitidos, en lugar de por tiempo de conexión. Además, los enlaces establecidos entre los distintos PSNs pueden compartirse entre varias

transferencias simultaneas, de forma multiplexada en el tiempo. No obstante, también ofrece alguna desventaja, especialmente en redes de tamaño pequeño porque la infraestructura de esta red es demasiado compleja y su coste bastante elevado.

Dentro de las redes de conmutación de paquetes se distinguen, a su vez, dos tipos:

- Redes de tipo datagrama: La característica principal de estas redes es que son redes poco fiables porque no garantizan la entrega correcta y ordenada de la información. Los paquetes pertenecientes a un mismo mensaje se envían de forma independiente unos de otros y pueden seguir caminos distintos. Los paquetes pueden llegar desordenados, duplicados, pueden sufrir errores en la transmisión o pueden perderse antes de alcanzar su destino. Todos los paquetes enviados deben contener la dirección de los extremos origen y destino.

  La ventaja principal de las redes de tipo datagrama en la sencillez de los nodos de conmutación, pues estos únicamente deben tomar decisiones de encaminamiento para cada paquete recibido, pero no tienen que preocuparse de controlar los errores por pérdidas o duplicidad de paquetes. Un ejemplo de red de tipo datagrama es la red Internet.

- Redes de tipo circuito virtual: Son redes fiables, que garantizan la entrega correcta y ordenada de la información. Los paquetes pertenecientes a un mismo mensaje siguen todos el mismo camino y estos siempre llegan a su destino ordenados, sin duplicados, sin errores y sin perdidas. En caso de que un paquete se pierda o sufra un error, los propios nodos de conmutación se encargan de su retransmisión.

  Antes de enviar datos es necesario establecer un camino o circuito virtual entre el origen y el destino a través de nodos de conmutación intermedios que se utilizara para enviar todos los paquetes del mensaje original. Los paquetes no necesitan llevar la dirección de las maquina origen y destino, basta con que lleven un identificador del camino o circuito virtual que deben seguir.

  La principal desventaja de las redes de circuito virtual es que los nodos de conmutación, además de realizar el encaminamiento de los paquetes, deben llevar a cabo un exhaustivo chequeo de errores para evitar pérdidas, duplicidad o desordenamiento de paquetes. Esto conlleva una complejidad adicional en el diseño de los nodos de conmutación que pueden redundar en un mayor conste del servicio, comparado con las redes de tipo datagrama. Un ejemplo de red de tipo circuito virtual es la red pública X.25.

1.2.1.4. Redes de difusión (broadcast) o de canal compartido

Las redes de difusión son una solución mucho más económica que las redes de conmutación de paquetes para la comunicación de datos en redes de medio o pequeño tamaño. En este tipo de redes, los computadores están unidos mediante un canal de comunicación compartido.

Cuando un computador quiere enviar información, la escribe en el canal compartido, de manera que todas las maquinas de la red pueden tener acceso a dicha información. Esta es la razón por la que reciben el nombre de redes de difusión, por que la información se difunde en el canal hacia todas las estaciones de la red.

En las redes de difusión, al igual que en las redes de conmutación de paquetes, la información se suele dividir en bloques, que en este contexto se denominan tramas. Estas tramas deben llevar la dirección de la maquina emisora y la destinataria. Solo la estación o estaciones destinatarias de una trama de datos realizan una copia de la misma. El resto de estaciones, aunque también pueden escuchar la trama circulando por el canal compartido, no la copian.

El principal inconveniente que presentan las redes de difusión es el problema de las colisiones. Este problema se plantea cuando dos o mas computadores quieren escribir simultáneamente sobre el canal compartido; entonces las tramas procedentes de los distintos equipos se solapan y la señal resultante es invalida.

Para resolver el problema de las colisiones se pueden emplear distintos mecanismos de control de acceso al medio. Existe una gran diversidad de redes de difusión que se diferencian en la topología que presentan (bus, anillo, estrella, etc.), el medio de transmisión empleado (par trenzado, cable coaxial, fibra óptica, etc.) y el mecanismo MAC para resolver las colisiones. La practica totalidad de redes de área local (Ethernet, Token Ring, Token Bus) y de área metropolitana (FDDI, etc) son redes de difusión.

1.2.2. Clasificación de las redes según el área geográfica abarcada

1.2.2.1. Redes de área local (LAN)

Las redes de área local (LAN, Local Area Networks) son redes que ocupan un área geográfica de tamaño moderado, que puede oscilar desde unos metros a unas centenas de metros. Suelen estar confinadas dentro de un mismo edificio o abarcar varios edificios colindantes.

Generalmente son totalmente privadas, pues pertenecen y son gestionadas por una única organización. Las redes LAN son generalmente redes de difusión o de canal compartido. Las redes de área local mas difundidas en la actualidad son de tipo Ethernet (10 Mbps), Fast Ethernet (100 Mbps) y Token Ring (16 Mbps).

1.2.2.2. Redes de área metropolitana (MAN)

Las redes de área metropolitana (MAN, Metropolitan Area Networks) ocupan un área geográfica más extensa, en torno a decenas de kilómetros y pueden abarcar un grupo amplio de edificios. Estas redes suelen pertenecer y estar gestionadas por una o varias organizaciones, ya sean privadas o públicas, que

habitualmente ofrecen servicios de conexión a otras empresas que utilizan la red MAN para interconectar sus redes LAN privadas.

Las redes MAN suelen ser también redes de difusión. La red de área metropolitana mas difundida en la actualidad es la red FDDI (Fiber Distributed Data Interface o Interfaz de datos distribuidos por fibra), una red de fibra óptica que transmite a 100 Mbps.

1.2.2.3. Redes de área extensa (WAN)

Las redes de área extensa (WAN, Wide Area Networks) abarcan distancias desde decenas de miles de kilómetros. Habitualmente interconectan unas ciudades con otras, e incluso unos países con otros. Este tipo de redes suelen emplear tecnología de redes de conmutación de paquetes y normalmente están gestionadas por algún operador de telecomunicaciones (publico o privado) que ofrece servicios de conexión de red. Uno de los tipos clásicos de redes de área extensa son las redes públicas de datos basadas en X.25.

1.3. Dispositivos de interconexión de redes

1.3.1. Repetidores

El repetidor es el dispositivo más simple capaz de interconectar dos o más segmentos de una red de área local, y actúa de manera que cualquier señal que recibe, la regenera, la amplifica y la retransmite, bit a bit, hacia todos los segmentos de salida. Los repetidores son, por tanto, dispositivos de la capa física que solo permiten interconectar segmentos de red idénticos.

Las redes 802.3 hacen uso muy frecuente de repetidores, pues los segmentos de red están limitados a una longitud máxima de 500 metros en la implementación 10BASE5 y 185 metros en la implementación 10BASE2. Si la extensión de la red abarca una longitud mayor, es necesario interconectar varios segmentos mediante repetidores.

El uso de repetidores en redes 802.3 presenta algunas limitaciones. En primer lugar no pueden existir más de cuatro repetidores en el camino entre dos estaciones cualesquiera. En segundo lugar, una red formada por varios segmentos, unidos por repetidores no puede contener lazos cerrados, porque de lo contrario cualquier información transmitida no dejaría nunca de circular por la red.

Aparte de estas limitaciones, el repetidor presenta una serie de ventajas y desventajas con respecto a otros dispositivos de interconexión más complejos. La principal ventaja de los repetidores es su simplicidad y rapidez, pues su única función es regenerar y retransmitir bit a bit la información recibida.

No tienen por tanto necesidad de de almacenar tramas de información o desencapsular las mismas para determinar su destino.

Por otra parte, la principal desventaja de los repetidores es la gran cantidad de tráfico que generan, pues no son capaces de filtrar el tráfico en función de su destino. En efecto, cuando un repetidor recibe una trama de datos la retransmite hacia el resto de salidas, independientemente de cual sea su destino específico. Este mecanismo se conoce como inundación y genera una gran cantidad de tráfico innecesario, pues todas las tramas llegan a todos los segmentos conectados a la red.

1.3.2. Conmutadores o Switches

Un switch es un dispositivo para interconexión de redes, capaz de proporcionar un camino de comunicación dedicado entre la red origen y a red destino.

El switch presenta dos ventajas principales con respecto al repetidor. En primer lugar, el switch reduce notablemente el tráfico en la red, puesto que es capaz de filtrar la información en función de la dirección física de la estación destinataria y solo reexpide los datos hacia la salida o salidas adecuadas. En segundo lugar, el switch permite establecer varios caminos de datos simultáneos entre distintas redes, por lo que aumenta el ancho de banda efectivo de la red.

Para poder reexpedir la información hacia el puerto de salida adecuado, el switch debe acceder a la trama MAC para leer la dirección de la estación destinataria. Se trata por tanto de un dispositivo que trabaja a nivel de capa MAC. A pesar de las dos importantes ventajas que ofrece, el switch es un dispositivo más lento que el repetidor, pues debe desencapsular al menos parcialmente las tramas de datos para conocer la dirección de destino.

Para poder reexpedir las tramas hacia las salidas adecuadas, el switch debe disponer de algún mecanismo que le permita conocer las direcciones de las estaciones que están conectadas a cada puerto. Para ello el switch dispone de una tabla que asocia direcciones físicas a puertos y, además, incorpora un mecanismo transparente de aprendizaje, de manera que cuando el switch recibe una trama de una estación a través de un determinado puerto, introduce en su tabla la dirección física de dicha estación asociada al puerto al que esta conectada.

Cuando un switch tiene que reexpedir una trama, en primer lugar consulta su tabla para conocer el puerto de salida adecuado. Si no dispone de esta información, entonces actúa como un repetidor y retransmite la trama por todas las salidas.

En función del mecanismo de reexpedición empleado se distinguen dos tipos básicos de switches:

- Switches de almacenamiento y reexpedición (store-and-forward): Este tipo de switches realiza una copia completa de cada trama que recibe y lleva a cabo la comprobación del código de redundancia (CRC), de manera que si la trama contiene errores es descartada. Si, por el contrario, la trama es correcta, se transmite hacia la salida adecuada en función de la dirección MAC destino. Este proceso

de almacenamiento de la trama, chequeo de errores y retransmisión de la trama, puede llegar a introducir retardos significativos en la red.

- Switches de truncamiento (cut-through): Este tipo de switches no copia totalmente cada trama que recibe, sino únicamente los primeros bytes hasta conocer la dirección de destino. Una vez determinada esta dirección, el resto de la trama se reexpide directamente hacia la salida adecuada. Estos switches no realizan chequeos de errores, pero a cambio los retardos introducidos son considerablemente menores que en el caso anterior.

El switch es un dispositivo muy empleado en la interconexión de redes Ethernet (10BASE-T) y Fast Ethernet (100-BASE-T). En este contexto podemos encontrar dos tipos de switches:

- Switches simétricos: En este tipo de switches todos los puertos trabajan a una misma velocidad, ya sea 10 Mbps o 100 Mbps, y por lo tanto no es posible conectar redes Ethernet y Fast Ethernet.
- Switches asimétricos: Los puertos de switches asimétricos pueden funcionar indistintamente a velocidades de 10 Mbps o 100 Mbps, por lo tanto también se denominan switches 10/100. Este tipo de switches permite mezclar redes Ethernet y Fast Ethernet. También hacen posible establecer varias conexiones simultáneas entre varias redes 10BASE-T y una misma red 100BASE-T.

Este tipo de interconexión resulta muy apropiado para un esquema de tipo cliente-servidor, en el cual las maquinas clientes se conectan a redes 10BASE-T y la maquina servidora a una red 100BASE-T. De esta forma, el servidor puede soportar un ancho de banda mayor y así atender simultáneamente a múltiples cliente sin problemas de saturación de la red.

Para poder intercambiar información entre redes de 10 Mbps y redes de 100 Mbps, el switch debe adaptar las velocidades de ambas redes. Para ello el switch tiene que almacenar cada trama que recibe y reexpedirla hacia la red destinataria a la velocidad adecuada de dicha red. Por tanto, han de ser switches de almacenamiento y reexpedición y, por lo general, deben disponer de una notable capacidad de almacenamiento.

1.3.3. Puentes o Bridges

Un puente es un dispositivo más general diseñado para interconectar dos o mas redes de área local, que pueden ser idénticas o distintas. Mientras que el repetidor actúa a nivel de la capa física y el switch a nivel de la capa MAC, el puente actúa a nivel de la capa LLC. Por tanto las redes que se interconectan mediante un puente pueden tener distinta capa MAC (802.3, 802.4, 802.5 u otras) pero idéntica capa LLC (802.2).

Un puente no reexpide la información bit a bit como un repetidor, sino que copia las tramas MAC completas, observa su dirección de destino y las redirige hacia las salidas o puertos adecuados. En cierto

modo, un puente es un dispositivo bastante parecido a un switch de almacenamiento y reexpedición, aunque presenta dos diferencias fundamentales.

En primer lugar, el puente puede interconectar redes de área local con diferente capa MAC y por tanto debe ser capaz de convertir los formatos de las tramas MAC de la red original a la red destino. En segundo lugar, los puentes pueden tomar decisiones de encaminamiento mas complejas que los switches, en caso de que exista más de una ruta posible entre la red origen y destino.

1.3.3.1. Interconexión de redes locales de distinto tipo mediante puentes

Cuando un puente interconecta redes LAN de distinto tipo. Debe realizar tres funciones básicas:

- Controlar la longitud de las tramas, pues las redes interconectadas pueden tener una longitud de trama distinta.
- Adaptar la diferencia de velocidad entre las distintas redes. Puesto que las LANs interconectadas pueden trabajar a velocidades distintas, el puente debe tener una cierta capacidad de almacenamiento para adaptar las velocidades de las distintas redes.
- Conversión de formatos de la trama MAC entre la red origen y la red destino.

1.3.3.2. Mecanismos de encaminamiento en puentes

En el caso mas general, podemos tener varias redes de area local, interconectadas por multiples puentes, de manera que para ir desde una estacion origen hasta una estacion destino sea necesario viajar a traves de varias LANs y varios puentes, cabiendo la posibilidad de que existan varias rutas alternativas entre un mismo origen y destino. Los puentes, por tanto, ademas de la traduccion del formato de las tramas, deben ser capaces de tomar decisiones de encaminamiento.

Este modo de operación exige que los puentes tengan conocimiento, por una parte, de que estaciones son accesibles a traves de ellos y, por otra parte, a que puerto de salida deben enviar la trama.

1.3.3.3. Puentes de encaminamiento estático

En los puentes de encaminamiento estático, las tablas de encaminamiento son fijas y se construyen e introducen en los puentes manualmente, antes de poner la red en funcionamiento. Cuando existen varias rutas alternativas entre dos LANs, se suele utilizar la ruta con menor número de saltos, aunque es posible usar otras métricas. Si dos rutas implican el mismo número de saltos, se elige una de ellas utilizando algún tipo de criterio, como puede ser, seleccionar el puente con identificador mas bajo.

La elaboración manual de tablas de rutas plantea un importante problema, porque si cambia la configuración de la red, las tablas de encaminamiento pueden resultar inadecuadas. Sería mucho mas

deseable que las tablas de encaminamiento se adaptasen de forma dinámica y automática a los cambios en la configuración de la red. Este es precisamente el modo de operación de los puentes transparentes.

1.3.3.4. Puentes transparentes

Los puentes trasparentes fueron diseñados inicialmente por DEC (Digital Equipment Corporation) y posteriormente fueron adoptados por el estándar IEEE 802.1. Los puentes transparentes permiten interconectar LANs que utilizan distintos protocolos MAC (802.3, 802.4 y 802.5), aunque son muy populares en la interconexión de redes 802.3/ Ethernet.

A diferencia de los puentes de encaminamiento estático, los puentes transparentes se caracterizan porque utilizan un mecanismo automático de aprendizaje para construir las tablas, que se basa en el análisis de la dirección fuente de las tramas que recibe por cada uno de sus puertos. Es decir, si un puente recibe una trama procedente de la estación N a través del puerto P, el puente automáticamente incluye en su tabla de encaminamiento que la estación N es accesible a través del puerto P. Cuando un puente tiene que reexpedir una trama, en primer lugar consulta su tabla para conocer el puerto de salida adecuado. Si no dispone de esa información, entonces retransmite la trama por todos los puertos (excepto por el que llegó).

Este camino de aprendizaje puede fallar cuando existen varios caminos entre dos LANs de la red que forman lazos cerrados.El problema de los lazos se puede resolver utilizando el Algoritmo del árbol de expansión o STA (Spaning-Tree Algorithm), que utiliza una de las conclusiones de la teoría de grafos para, a partir de una red con lazos, construir una red con una conectividad equivalente pero libre de lazos.

Este teorema dice lo siguiente: para cualquier grafo conexo consistente en una serie de nodos y una serie de arcos que conectan parejas de nodos, existe un árbol de expansión formado por arcos que mantienen la conectividad del grafo, pero que no contiene lazos.

El árbol de expansión, garantiza que entre una LAN origen y una LAN destino solo existe un camino posible y se calcula cada vez que se conecta un nuevo puente a la red, o cada vez que se detecta un cambio en la topología. El cálculo del árbol de expansión requiere un intercambio de información entre todos los puentes de la red y el puente raíz. Este intercambio de información se lleva acabo mediante unas tramas especiales de configuración, denominadas unidades de datos del protocolo del puente o BPDUs (Bridge Protocol Data Units).

1.3.3.5. Puentes de encaminamiento de origen

Los puentes de encaminamiento de origen fueron propuestos inicialmente por IBM, y posteriormente adoptados por IEEE formando parte del estándar 802.5.

Originalmente, los puentes de encaminamiento de origen se desarrollaron para la interconexión de LANs de tipo Token Ring, aunque también se pueden utilizar para la interconexión de LANs con distintos

protocolos MAC. La filosofía de los puentes de encaminamiento de origen consiste en que una estación emisora debe determinar con antelación la ruta que debe seguir cada trama.

Esta información de encaminamiento debe estar contenida en la propia trama, de manera que cuando un puente recibe una trama puede leer su información de encaminamiento y decidir si debe reexpedirla o desecharla.Si una estación quiere enviar una trama a otra estación ubicada en una LAN distinta, inicialmente la estación emisora desconocerá la ruta que debe seguir la trama para alcanzar la estación de destino.

Para determinar dicha ruta, la estación emisora envía una trama de exploración. Cada puente que recibe una trama de exploración, reexpide una copia de la misma por cada una de sus salidas (excepto por la que llego), añadiendo previamente a cada copia la información de ruta necesaria, es decir , el identificador del puente y el identificador de la red a la que se reexpide.

Cuando la estación destinataria recibe las tramas de exploración (puede recibir mas de una copia), responde a cada una de ellas de forma individual, utilizando la ruta acumulada en cada trama de exploración, de forma invertida. Cuando la estación emisora recibe estas respuestas, selecciona una de las rutas. El estándar 802.5 no especifica que criterios se debe seguir para seleccionar una ruta entre varias posibles, aunque los criterios mas habituales son seleccionar la ruta de la primera respuesta recibida o seleccionar la ruta mas corta (se pueden usar distintas métricas).

Una vez seleccionada una ruta, la estación emisora incluye dicha ruta en todas las tramas que vayan dirigidas a ese destino. Esta información se inserta en un campo denominado Campo de información de ruta o RIF (Routing Information Field). Cuando se utiliza el mecanismo de encaminamiento de origen, la presencia del campo de información de ruta (RIF) dentro de una trama se indica poniendo a 1 el primer bit del campo de dirección MAC origen, que es denominado bit indicador de información de ruta o RII (Routing Information Indicator).

Si el bit RII esta a 1, a continuación de la dirección MAC origen, se incluye la ruta que debe seguir la trama. Esta ruta consiste en una secuencia alternante de identificadores de LANs e identificadores de puentes, que debe comenzar y terminar con sendos identificadores de LAN (LAN origen y LAN destino, respectivamente).

1.3.3.6. Puentes transparentes de encaminamiento de origen

Aunque, en teoría , tanto los puentes transparentes como los puentes de encaminamiento de origen se pueden utilizar para interconectar cualquier tipo de red LAN, en la practica, la mayoría de puentes transparentes solo puede utilizarse para interconectar LANs de tipo 802.3 y 802.4 y la mayoría de los puentes de encaminamiento de origen solo pueden utilizarse para interconectar LANs de tipo 802.5. Como ambos tipos de puentes son totalmente incompatibles, resulta muy difícil interconectar LANs 802.3 y 802.4 con LANs 802.5.

La solución a esta incompatibilidad son los puentes transparentes de encaminamiento de origen. Estos

puentes pueden actuar bien como un puente transparente o bien como un puente de encaminamiento de origen, según el formato de la trama que reciban. Para ello, estos puentes chequean el primer bit de la dirección MAC origen de la trama. Este bit, denominado RII, solo se utiliza en puentes de encaminamiento de origen. Por tanto, si RII=1, el puente se comporta como un puente de encaminamiento de origen y si RII=0, el puente se comporta como un puente transparente.

1.3.4. Encaminadores o Routers

La utilización de puentes esta limitada a la interconexión de redes de área local que tienen en común la capa LLC. En un caso más general, para conectar una red de área local a otros tipos de redes (redes de área extensa, redes publicas X.25, etc) es necesario utilizar un dispositivo denominado encaminador o router, que trabaja a nivel de red.

Para poder interconectar redes a través de un router, es imprescindible que todas las redes, independientemente del tipo que sean, compartan el mismo protocolo de la capa de red y superiores. En la actualidad, el protocolo de red mas extendido es el protocolo IP (Internet Protocol). Este es el protocolo de red utilizado en Internet, que es una interconexión de redes de alcance mundial.

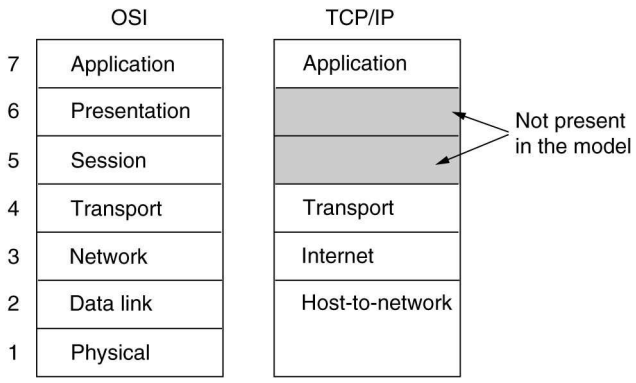
Un router de la red Internet es un dispositivo que trabaja a nivel IP y que tiene básicamente dos misiones:

- Realizar la traducción de protocolos de los niveles inferiores entre la red origen y la red destino.
- Proporcionar los mecanismos de encaminamiento necesarios para alcanzar cualquier estación destino desde cualquier estación origen, ambas conectadas a Internet, a través de redes y routers intermedios.

1.4. Modelo OSI

El modelo OSI (Open System Interconnection o interconexión de sistemas abiertos) fue publicado por la organización ISO (Internacional Standards Organization) como un marco de referencia para la definición de estándares para redes de comunicación de paquetes. En realidad, el modelo OSI no constituye un estándar de red por sí mismo, sino que define un esquema de arquitectura de red basado en siete capas y determina la funcionalidad básica de cada una de estas capas, así como el conjunto de primitivas de servicio elementales que debe ofrecer cada capa a la inmediatamente superior.

Figure 1-1. Pila OSI



A continuación se describen las principales funciones de cada uno de los niveles del modelo OSI.

1.4.1. Capa física

La capa física se ocupa de la transmisión de un flujo de bits a lo largo de un canal de comunicación real. La capa física define los parámetros básicos de la comunicación tales como:

- El medio de transmisión empleado: par trenzado, cable coaxial, fibra óptica, infrarrojos, microondas, etc.
- El tipo de conector usado para conectar el computador a la red, así como la función que desempeña cada una de las líneas de ese conector.
- El tipo de señales empleadas en la transmisión: analógicas o digitales.
- Los niveles eléctricos utilizados para codificar el 1 lógico y el 0 lógico.
- La velocidad de transmisión o, lo que es lo mismo, el intervalo de duración de cada bit.

1.4.2. Capa de enlace

La principal tarea de la capa de enlace será convertir el enlace físico de comunicación entre cada pareja de nodos de la red, propenso a errores de transmisión, en un enlace lógico fiable y libre de errores. Para ello la capa de enlace deberá ser capaz de realzar las siguientes funciones:

- Entramado de la información. La capa de enlace deberá encapsular la información procedente de la capa superior en tramas de datos con un formato específico. La función de entramado deberá habilitar

un mecanismo que permita al extremo receptor determinar dónde comienza y donde termina cada trama, por ejemplo incluyendo patrones especiales de principio y fin de trama.

- Detección de errores. La capa de enlace debe incluir mecanismos de detección de errores que permitan determinar al extremo receptor si una trama es correcta o si por el contrario, ha sufrido alguna alteración durante la transmisión debido a ruidos y otros fenómenos adversos. La detección de errores suelen llevarse a cabo mediante la inclusión de códigos de paridad o códigos de redundancia cíclica (CRC) en cada trama transmitida.
- Recuperación ante fallos. La capa de enlace se encarga de la retransmisión de todas las tramas perdidas o erróneas cuado se produce un fallo en la transmisión. El mecanismo más habitual consiste en que el extremo receptor envía una confirmación (ACK, Acknowledge) por cada trama correcta recibida. Si el emisor no recibe la trama de confirmación, significa que la trama de datos correspondiente se ha perdido o ha sufrido un error en la transmisión, por lo que deberá retransmitirla.
- Control de flujo. Para evitar la pérdida de daros es necesario evitar que un emisor rápido pueda llegar a saturar a un receptor lento. Para ello los mecanismos de control de flujo permiten regular la cantidad de información que se puede enviar al extremo receptor en función de las limitaciones de éste.

1.4.3. Capa de red

La capa de red tiene como tarea fundamental proporcionar un mecanismo de encaminamiento o routing entre los dos extremos de la comunicación a través de una serie de nodos de conmutación intermedios. Si el servicio de red es de tipo datagrama (sin conexión) la capa de red deberá ser capaz de encaminar uno por uno y de forma independiente cada uno de los paquetes que forman un mensaje original.

Si, por el contrario, la red ofrece un servicio de circuito virtual (orientado a conexión) la capa de red deberá determinar el camino más adecuado entre el origen y el destino para todos los paquetes que forman el mensaje.

Además de la función de encaminamiento, la capa de red también debe realizar el control de la congestión, que consiste en evitar que determinados puntos de la red puedan llegar a saturarse por causa de un tráfico excesivo, pues esto puede provocar una pérdida de información.

En efecto, si un nodo de conmutación recibe paquetes a un ritmo superior al que puede reexpedirlos, éste puede llegar a saturarse y entonces podrá rechazar todos los paquetes recibidos. Para evitar este problema será necesario distribuir adecuadamente la carga de la red buscando caminos alternativos para eludir las rutas más saturadas.

1.4.4. Capa de transporte

La principal función de la capa de transporte es proporcionar un servicio fiable extremo a extremo. En una red de conmutación de paquetes, el hecho deque cada uno de los enlaces físicos entre cada pareja de

nodos de la red sea fiable no significa que la información llegue correctamente a su destino, pues un paquete de datos puede perderse por el camino si, por ejemplo, un nodo de conmutación se satura.

La capa de transporte deberá resolver estas posibles situaciones de pérdida de datos, para lo que deberá disponer de u mecanismo de resuperación ante fallos similar a los empleados en a capa de enlace pero, en este caso, aplicados a los extremos de la comunicación.

Además de esta función, la capa de transporte también se encarga de dividir los datos procedentes de la capa superior en fragmentos de un tamaño admisible por la red. El extremo receptor deberá ser capaz de reensamblar los distintos fragmentos para formar el mensaje original, por lo que será necesario disponer de un mecanismo de numeración de los segmentos para saber el orden que ocupan dentro del mensaje.

1.4.5. Capa de sesión

La capa de sesión tiene como objetivo principal organizar y controlar la relación de comunicación o diálogo entre los dos extremos de la conexión. De esta forma, la capa de sesión debe ser capaz de establecer y mantener un canal de comunicación activo entre ambos equipos mientras dura el intercambio de datos, y liberar dicho canal cuando finaliza la transacción de información. Además de la tarea del control del diálogo, la capa de sesión puede realizar las siguientes funciones:

- Control del derecho de transmisión. Cuando se establece un diálogo entre dos extremos, se puede realizar una conexión full-duplex; que significa que ambos extremos pueden transmitir o recibir de forma simultánea o una conexión half-duplex; que significa que ambos extremos pueden transmitir o recibir, pero no de forma simultánea sino alternada. En este último caso, la capa de sesión proporciona los mecanismos necesarios para controlar el intercambio de datos y conceder el derecho de transmisión a uno u otro extremo de forma adecuada.
- Sincronización de la comunicación. Este servicio de la capa de sesión se encarga de devolver los extremos de la conexión a un estado conocido cuado se produce algún error o desacuerdo. Para ello la capa de sesión inserta periódicamente puntos de sincronización entre los datos transmitidos, de manera que si ocurre un fallo durante una transmisión de larga duración, el diálogo puede reiniciarse a partir del último punto de sincronización, en lugar de reiniciarse desde el principio.

1.4.6. Capa de presentación

El objetivo principal de la capa de presentación es ocuparse de los aspectos relacionados con la representación y la semántica de los datos transmitidos. Puesto que los datos pueden representarse de forma distinta en cada máquina (los enteros pueden representarse, por ejemplo, en complemento a 1 o complemento a 2 entre otros, los caracteres como ASCII o EBCDIC, etc.) es necesario establecer un formato de representación intermedio común a todas las máquinas de la red.

De esta forma, la capa de representación del extremo emisor debe realizar la transformación de la representación interna de la máquina a la representación intermedia en la red y el extremo receptor debe realizar la conversión contraria. Además de esta tarea, la capa de presentación puede realizar las siguientes funciones opcionales:

- Comprensión y descompresión de los datos para reducir el tiempo de transmisión.
- Cifrado y descifrado de la información mediante técnicas de criptografía para garantizar la integridad y privacidad de los datos transmitidos.

1.4.7. Capa de aplicación

La función principal de la capa de aplicación es interactuar con el usuario final proporcionando una interfaz de usuario formada por una amplia variedad de servicios o aplicaciones de la red. Algunas de las aplicaciones más comunes que constituyen esta capa son las siguientes:

- Transferencia de ficheros.
- Correo electrónico.
- Ejecución remota de trabajos.
- Servicio de directorio.
- Servicio de terminal virtual.
- Acceso a bases de datos remotas.

Dentro de la capa de aplicación es importante destacar tres servicios: DNS, Correo y Web.

1.4.7.1. DNS

El Domain Name System (DNS) es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Aunque como base de datos el DNS es capaz de asociar distintos tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio.

La asignación de nombres a direcciones IP es ciertamente la función más conocida de los protocolos DNS. Además de ser más fácil de recordar, el nombre es más fiable. La dirección numérica podría cambiar por muchas razones, sin que tenga que cambiar el nombre.

Inicialmente, el DNS nació de la necesidad de recordar fácilmente los nombres de todos los servidores conectados a Internet. En un inicio, SRI (ahora SRI International) alojaba un archivo llamado HOSTS que contenía todos los nombres de dominio conocidos.

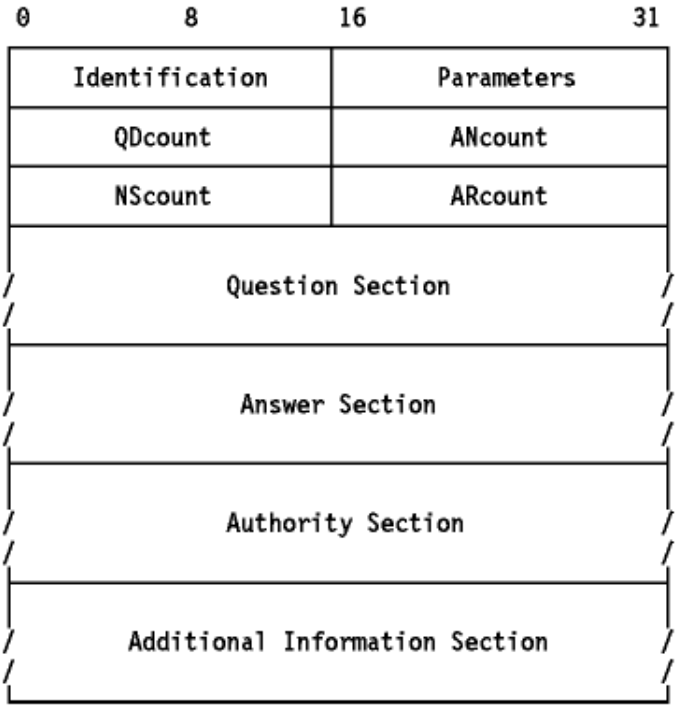
El crecimiento explosivo de la red causó que el sistema de nombres centralizado en el archivo HOSTS no resultara práctico y en 1983, Paul Mockapetris publicó los RFCs 882 y 883 definiendo lo que hoy en día ha evolucionado al DNS moderno. (Estos RFCs han quedado obsoletos por la publicación en 1987 de los RFCs 1034 y 1035).

Componentes DNS:

- Los Clientes DNS (resolvers), un programa cliente DNS que se ejecuta en la computadora del usuario y que genera peticiones DNS de resolución de nombres a un servidor DNS.
- Los Servidores DNS (name servers), que contestan las peticiones de los clientes, tienen la capacidad de reenviar la petición a otro servidor si no disponen de la dirección solicitada.
- Las Zonas de autoridad, porciones del espacio de nombres de dominio que almacenan los datos. Cada zona de autoridad abarca al menos un dominio y posiblemente sus subdominios, si estos últimos no son delegados a otras zonas de autoridad.

Todos los mensajes del DNS utilizan un único formato que se muestra a continuación. El "resolver" envía la trama al servidor de nombres. Sólo la cabecera y la sección "question" se utilizan para la consulta. Las respuestas o retransmisiones de las consultas usan la misma trama, pero llenan más secciones de la misma (las secciones "answer/authority/additional").

Figure 1-2. Formato del mensaje DNS.



Tipos de registros DNS:

- A: Address (Dirección) Este registro se usa para traducir nombres de hosts a direcciones IPv4 de 32 bits.
- AAAA: Address (Dirección) Este registro se usa para traducir nombres de hosts a direcciones IPv6 de 128 bits.
- CNAME: Canonical Name (Nombre Canónico) Se usa para crear nombres de hosts adicionales, o alias, para los hosts de un dominio.
- NS: Name Server (Servidor de Nombres) Define la asociación que existe entre un nombre de dominio y los servidores de nombres que almacenan la información de dicho dominio. Cada dominio se puede asociar a una cantidad cualquiera de servidores de nombres.
- MX: Mail Exchange (Intercambiador de Correo) Define el lugar donde se aloja el correo que recibe el dominio.
- PTR: Pointer (Indicador) También conocido como "registro inverso", funciona a la inversa del registro A, traduciendo IPs en nombres de dominio.
- SOA: Start of authority (Autoridad de la zona) Proporciona información sobre la zona.
- HINFO: Host INFOrmation (Información del sistema informático) Descripción del host, permite que la gente conozca el tipo de máquina y sistema operativo al que corresponde un dominio.
- TXT: TeXT (Información textual) Permite a los dominios identificarse de modos arbitrarios.

1.4.7.2. Web

Un servidor Web es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para lo que llamamos hipertextos, páginas web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. Sin embargo, el hecho de que HTTP y HTML estén íntimamente ligados no debe dar lugar a confundir ambos términos.

HTML es un lenguaje de programación y un formato de archivo y HTTP es un protocolo. Cabe destacar el hecho de que la palabra servidor identifica tanto al programa como a la máquina en la que dicho programa se ejecuta. Un servidor web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita.

A modo de ejemplo, al teclear http://www.ucm.es en nuestro navegador se produce una secuencia de tres pasos:

1. El navegador realiza una petición HTTP al servidor de dicha dirección, en este caso 147.96.1.15.
2. El servidor responde al cliente enviando el código HTML de la página.
3. El cliente, una vez recibido el código, lo interpreta y lo muestra en pantalla.

Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página. El servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

1.4.7.3. Correo

Un servidor de correo es una aplicación que nos permite enviar mensajes (correos) de unos usuarios a otros, con independencia de la red que dichos usuarios estén utilizando. Para lograrlo se definen una serie de protocolos, cada uno con una finalidad concreta:

- SMTP, Simple Mail Transfer Protocol: Es el protocolo que se utiliza para que dos servidores de correo intercambien mensajes. SMTP se basa en el modelo cliente-servidor, donde un cliente envía un mensaje a uno o varios receptores. La comunicación entre el cliente y el servidor consiste enteramente en líneas de texto compuestas por caracteres ASCII. El tamaño máximo permitido para estas líneas es de 1000 caracteres.

  Las respuestas del servidor constan de un código numérico de tres digitos, seguido de un texto explicativo. El número va dirigido a un procesado automático de la respuesta por autómata, mientras que el texto permite que un humano interprete la respuesta. Además, todas las réplicas tienen un código numérico al comienzo de la línea.

  El mensaje está compuesto por dos partes:

  - Cabecera: Usa unas palabras clave para definir los campos del mensaje. Éstos campos ayudan a los clientes de correo a organizarlos y mostrarlos. Los más típicos son subject (asunto), from (emisor) y to (receptor). Éstos dos últimos campos no hay que confundirlos con las órdenes MAIL FROM y RCPT TO, que pertenecen al protocolo, pero no al formato del mensaje.
  - Cuerpo del mensaje: Es el mensaje propiamente dicho. En el SMTP básico está compuesto únicamente por texto, y finalizado con una línea en la que el único carácter es un punto.

- POP, Post Office Protocol: Se utiliza para obtener los mensajes guardados en el servidor y pasárselos al usuario. El diseño de POP3 y sus predecesores permite que los usuarios con conexiones intermitentes (tales como las conexiones módem), descarguen su correo-e cuando se encuentren conectados de tal manera que puedan ver y manipular sus mensajes sin necesidad de permanecer conectado.

  Cabe mencionar que la mayoría de los clientes de correo incluyen la opción de dejar los mensajes en el servidor, de manera tal que, un cliente que utilice POP3 se conecta, obtiene todos los mensajes, los almacena en la computadora del usuario como mensajes nuevos, los elimina del servidor y finalmente se desconecta.

- IMAP, Internet Message Access Protocol: Su finalidad es la misma que la de POP, pero el funcionamiento y las funcionalidades que ofrecen son diferentes. IMAP fue diseñado como una

moderna alternativa a POP por Mark Crispin en el año 1986. IMAP es utilizado frecuentemente en redes grandes, por ejemplo los sistemas de correo de un campus.

IMAP le permite a los usuarios acceder a los nuevos mensajes instantáneamente en sus computadoras, ya que el correo está almacenado en la red. Con POP3 los usuarios tendrían que descargar el email a sus computadoras o accederlo vía web. Ambos métodos toman más tiempo de lo que le tomaría a IMAP, y se tiene que descargar el email nuevo o refrescar la página para ver los nuevos mensajes. De manera contraria a otros protocolos de Internet, IMAP4 soporta mecanismos nativos de cifrado. La transmisión de contraseñas en texto plano también es soportada.

Así pues, un servidor de correo consta en realidad de dos servidores: un servidor SMTP que será el encargado de enviar y recibir mensajes, y un servidor POP/IMAP que será el que permita a los usuarios obtener sus mensajes. Para obtener los mensajes del servidor, los usuarios se sirven de clientes, es decir, programas que implementan un protocolo POP/IMAP. En algunas ocasiones el cliente se ejecuta en la máquina del usuario

1.4.8. Flujo de información en la arquitectura OSI

En una arquitectura organizada en niveles como establece el modelo OSI, la información que proporciona el usuario a la capa de aplicación va pasando de un nivel a otro hasta alcanzar el nivel físico, donde la información es tratada como un simple flujo de bits que se transmite al extremo contrario. Cada una de estas capas incluye una cabecera de control (y opcionalmente una cola) necesaria para llevar a cabo las funciones propias del protocolo asociado a dicha capa.

1.4.9. El modelo de referencia para redes de área local

El modelo OSI se diseñó inicialmente como un marco de referencia para la definición de estándares de redes de conmutación de paquetes. Sin embargo, las redes de área local y metropolitanas, al ser redes de difusión, presentan una problemática diferente que no se da en las redes de conmutación de paquetes, se trata del problema de las colisiones.

Para resolver este problema es necesario incluir una capa adicional que gestione el control de acceso al canal de comunicación compartido, de manera que se eviten las colisiones o, en caso de que se produzcan, éstas se puedan detectar y resolver. Esta capa se denomina capa MAC (Medium Access Control o control de acceso al medio) y está ubicada entre la capa física y la capa de enlace. En este modelo se distinguen tres capas:

- Capa física. Define la naturaleza del medio de transmisión, los detalles de la conexión del dispositivo a la red, las características de las señales eléctricas, etc.
- Capa MAC (Medium Access Control o control de acceso al medio). Define el protocolo utilizado para controlar el acceso al canal de comunicaciones compartido y evitar o resolver el problema de las colisiones.

- Capa LLC (Logic Link Control o control lógico del enlace). Representa la capa de enlace de las redes de área local y define el modo de establecer, mantener y terminar un enlace lógico entre dispositivos.

Como puede observarse, el modelo de referencia de redes de área local no incluye la descripción de las capas de nivel superior. Esto es debido a que este modelo de referencia pretende ser un modelo abierto compatible con distintas arquitecturas de protocolos de alto nivel.

Table 1-1. Tecnologías y protocolos de red organizados por niveles OSI

Nivel	Protocolos o tecnologías
Nivel de aplicación	DNS, FTP, HTTP, IMAP, IRC, NFS, NNTP, NTP, POP3, SMB/CIFS, SMTP, SNMP, SSH, Telnet, SIP...
Nivel de presentación	ASN.1, MIME, SSL/TLS, XML, XDR...
Nivel de sesión	NetBIOS, ONC RPC, DCE/RPC, XML-RPC...
Nivel de transporte	SCTP, SPX, TCP, UDP...
Nivel de red	AppleTalk, IP, IPX, NetBEUI, X.25...
Nivel de enlace	ATM, Ethernet, Frame Relay, HDLC, PPP, Token Ring, Wi-Fi, STP...
Nivel físico	Cable coaxial, Cable de fibra óptica, Cable de par trenzado, Microondas, Radio, RS-232...

Chapter 2. El protocolo IP de Internet

2.1. IPv4

IP es un protocolo ideado para interconexión de redes heterogéneas mediante routers. Se trata de un protocolo de conexión no fiable, que no garantiza la entrega segura de los paquetes. Además, los paquetes que se transmiten a la red, aunque pertenezcan a un mismo mensaje original, pueden seguir caminos diferentes, por lo que pueden llegar desordenados e incluso duplicados. Deberá ser la capa de transporte, o incluso la propia aplicación, la que, en su caso, detecte y resuelva todas estas situaciones de error. Las unidades de información que se transmiten a nivel del protocolo IP se denominan paquetes IP o datagramas.

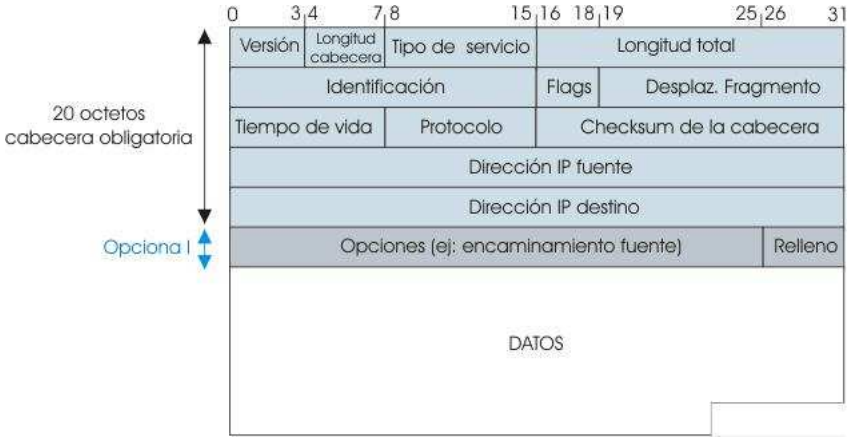
El protocolo IP tiene tres funciones básicas:

- Direccionamiento. IP debe proporcionar un conjunto global de direcciones que permitan identificar de forma unívoca a cada una de las máquinas conectadas a Internet. Estas direcciones se conocen con el nombre de direcciones IP y no deben confundirse con las direcciones físicas o MAC que se utilizan a nivel de la capa de control de acceso al medio en redes de área local.
- Encaminamiento. IP debe incorporar mecanismos de encaminamiento eficientes que permitan a todas las estaciones y routers de Internet encaminar correctamente los datagramas en función de su destino. Para poder llevar a cabo todas estas funciones, todos los datagramas que se transmiten a la red deben incluir las direcciones IP de las maquinas origen y destino.
- Fragmentación. Cuando un datagrama tiene que cruzar a través de una o varias redes en el camino hacia su destino, el protocolo IP debe encargarse de dividir el paquete en fragmentos de un tamaño aceptable por cada una de las redes que atraviesa. Igualmente, en el destino, el protocolo IP debe ser capaz de reensamblar los distintos fragmentos recibidos para formar el datagrama original.

2.1.1. Formato del datagrama IPv4

Un datagrama IP está formado por una cabecera IP y una zona de datos. La cabecera tiene un tamaño mínimo de 20 bytes y está formada por palabras de 32 bits (5 palabras como mínimo).

Figure 2-1. Esquema de un paquete IPv4



A continuación se detallan los principales campos de la cabecera IPv4:

- Campo Versión: Determina la versión del protocolo IP al que pertenece el datagrama (actualmente se utilizan las versiones 4 y 6).
- Campo IHL (Internet Header Length): Longitud de la cabecera medida en palabras de 32 bits (la longitud mínima son 5 palabras).
- Campo Tipo de servicio. Permite especificar determinados parámetros de calidad del servicio, tales como prioridad del datagrama, la rapidez de la entrega, la seguridad en la entrega, etc. La gran parte de los routers existentes en la actualidad no son capaces de procesar esta información, por lo que en la mayoría de las ocasiones el hecho de especificar parámetros de calidad del servicio no supone un tratamiento especial para el datagrama.
- Campo Longitud total: Indica la longitud total del datagrama (medida en bytes), incluyendo la cabecera y la zona de datos. Dado que este campo es de 16 bits, la longitud máxima del datagrama podrá ser de 64 Kbytes.
- Campo Identificación: Permite llevar a cabo la fragmentación de los datagramas, junto con los campos MF, DF y Desplazamiento. Todos los datagramas llevan un identificador, de manera que cuando un datagrama se fragmenta, a cada uno de los fragmentos se le asigna el mismo identificador.
- Campo MF (More Fragments): Si vale 0, significa que se trata del último fragmento de un datagrama. Si vale 1, indica que hay más fragmentos del datagrama.
- Campo DF (Don’t Fragment): Si vale1, significa que el datagrama no se puede fragmentar. Si vale 0 indica que el datagrama si se puede fragmentar.
- Campo Desplazamiento: Indica la posición que ocupa el fragmento dentro del datagrama original. Está campo contiene la distancia, medida en bloques de 8 bytes, desde el inicio del fragmento hasta el inicio del datagrama original. Cuando se fragmenta un datagrama, todos los fragmento, excepto quizás el último, deben contener un número entero de bloques.
- Campo Tiempo de vida: Contiene el tiempo máximo, normalmente medido en número de saltos, que el datagrama puede permanecer circulando por la red. Cada vez que el datagrama pasa a través de un

router, este le resta 1 al tiempo de vida. Cuando el campo llega a 0, el router descarta el datagrama.

- Campo Protocolo: Identifica el protocolo de la capa superior al que pertenecen los datos. La capa IP se comunica básicamente con dos protocolos: TCP (campo protocolo = 6) y UDP (campo protocolo = 17).
- Campo Código de redundancia de la cabecera: Se utiliza para la detección de errores en la cabecera del datagrama. Este campo se calcula realizando la operación lógica O-exclusiva (XOR) de todas las palabras de 16 bits que forman la cabecera.
- Direcciones IP fuente y destino: Estas direcciones identifican la máquina emisora y receptora del datagrama. Cada estación conectada a Internet debe tener una dirección IP única que no se puede repetir en ninguna otra estación de la red.
- Campo Opciones: Incluye información adicional y opcional del datagrama. La información que suele contener es del siguiente tipo: Encaminamiento en origen, Registro de ruta y Sello de tiempo.

2.1.2. Direcciones IP

Las direcciones IP (fuente y destino) están formadas por dos partes: una parte de la dirección identifica a la red y otra parte identifica a la estación dentro de la red. La parte que identifica a la red es fija para cada red y en necesario contratarla a la organización NIC (Network Information Center), que gestiona a nivel mundial la asignación de direcciones IP. La parte que identifica al host puede asignarla libremente el administrador de la red a los host de su red. Las direcciones IP constan de 32 bits, es decir, 4 bytes. Para referirnos a las direcciones IP se suele utilizar la denominada notación de punto, que consiste en expresar cada uno de los bytes mediante su valor decimal equivalente y separar dichos bytes unos de otros mediante puntos. Por ejemplo: 11000110 00100010 01000010 00001010 = 198.34.66.10 Las direcciones IP pueden ser de 5 tipos o clases, cuyo formato se muestra en la figura. Estas clases son las siguientes:

- Direcciones de clase A: Están reservadas para redes de tamaño muy grande: hasta 126 redes de aproximadamente 16 millones de estaciones cada una. El primer byte es fijo y asignado por el NIC. Este primer byte toma un valor entre 0 y 127, aunque los valores extremos, 0 y 127, no se usan como direcciones reales de máquinas. Los tres últimos bytes puede asignarlos libremente el administrador de la red a las distintas estaciones que la componen.
- Direcciones de clase B: Están reservadas para redes de tamaño medio-grande: 16384 redes con aproximadamente 65000 estaciones cada una. Los dos primeros bytes son fijos y están asignados por el NIC. En este tipo de redes, el primer byte toma un valor entre 128 y 191. Los dos últimos bytes puede asignarlos libremente el administrador de la red a las distintas estaciones que la componen.
- Direcciones de clase C: Se utilizan para redes de pequeño tamaño: aproximadamente 2 millones de redes con 254 estaciones cada una. Los tres primeros bytes son fijos y están asignados por el NIC. En este tipo de redes, el primer byte toma un valor entre 192 y 233. El último byte puede asignarlo libremente el administrador de la red a las distintas estaciones que la componen.
- Direcciones de la clase D: Las direcciones de la clase D también se denominan direcciones multicast o direcciones de grupo, de manera que una misma dirección hace referencia a un grupo de direcciones repartidas a lo largo de Internet. Cuando se envía un paquete a una dirección multicast, este llegará a

todas las máquinas que forman parte de dicho grupo. El primer byte de las direcciones multicast toma siempre un valor entre 224 y 239.

- Direccion de la clase E: Las direcciones de la clase E están reservadas para uso experimental. El primer byte de las direcciones de la clase toma un valor entre 240 y 254.

2.1.3. Direcciones IP especiales

De todas las direcciones indicadas anteriormente, existe una serie de direcciones especiales que no se asignan a ninguna máquina de Internet. Estas direcciones son las siguientes:

- Direcciones de loopback o bucle interno (127.x.y.z): Estas direcciones no se asignan a ninguna red y se utilizan para el bucle interno (loopback), que permite a una maquina aislada ejecutar aplicaciones de red.
- Dirección desconocida (0.0.0.0): Esta dirección no se asigna a ninguna red. Esta es la dirección que utiliza el protocolo RARP o protocolo de resolución inversa de direcciones cuando una estación quiere averiguar su propia dirección IP a partir de su dirección MAC.
- Dirección broadcast o de difusión (terminadas en 255): Las direcciones broadcast se utilizan para enviar un paquete a todas las máquinas de la red. La dirección broadcast universal es 255.255.255.255.
- Dirección de red (terminadas en 0): Para referirnos a una red completa se usa la dirección de la red terminada en ceros. Este número se denomina dirección oficial de la red. Las direcciones oficiales de red se utilizan en las tablas de encaminamiento, para decidir hacia donde debemos enviar un paquete en función de la red destinataria.

2.1.4. Máscaras de red y tablas de encaminamiento

Las máscaras de red, junto con las tablas de encaminamiento, se utilizan en una estación para decidir hacia donde se debe redirigir un paquete, en función de su destino. El valor de la máscara de red varía en función de la clase de la red. Se trata de un número de 4 bytes que toma los siguientes valores:

- Red de clase A: máscara de red = 255.0.0.0
- Red de clase B: máscara de red = 255.255.0.0
- Red de clase C: máscara de red = 255.255.255.0

2.1.5. Protocolo ARP

Cuando el usuario utiliza una aplicación de Internet para comunicarse con una estación remota emplea la dirección IP de dicha estación. Para poder transmitir un paquete a la estación destinataria a través de la

red local (o al router, si la estación destinataria está en una red distinta) es necesario encapsular dicho paquete dentro de una trama MAC y por tanto, es imprescindible conocer la dirección física de la estación destinataria. En otras palabras, a nivel de aplicación se utilizan direcciones IP, pero a nivel de comunicación física se utilizan direcciones MAC y en principio, no existe ninguna relación entre la dirección IP de una máquina y su dirección MAC. Es necesario, por tanto, realizar una traducción de la dirección IP a la dirección MAC. Esta traducción se realiza mediante el protocolo ARP (Address Resolution Protocol o protocolo de resolución de direcciones).

El protocolo ARP consta de una tabla de traducción de direcciones, denominada tabla ARP, donde están ubicadas las direcciones IP de las últimas estaciones con las que se ha comunicado recientemente y las direcciones MAC asociadas. Cuando una estación A se quiere comunicar con otra estación B, en primer lugar la estación A busca en la tabal ARP la dirección MAC de B. Si la dirección MAC de B no esta en la tabla ARP, entonces entra en funcionamiento el protocolo ARP. Este protocolo consiste en que la estación A envía un mensaje broadcast dirigido a todas las estaciones de la red, que contiene la dirección IP de la estación B. La estación B, cuando reconoce su dirección en el mensaje broadcast, envía un mensaje de respuesta dirigido a la estación A informando a ésta de cual es su dirección MAC.

2.1.6. Protocolo RARP

Existen ocasiones en que una máquina únicamente conoce su dirección Ethernet, pero desconoce su dirección IP. El ejemplo más típico son estaciones sin disco, que requieren de la presencia de un servidor que les proporcione una copia del sistema operativo a través de la red para poder arrancar adecuadamente. Cuando una estación sin disco arranca, lo primero que necesita es que otra máquina le proporcione una dirección IP. Esto se lleva a cabo mediante el protocolo RARP (Reverse Address Resolution Protocol o protocolo de resolución inversa de direcciones). El protocolo RARP funciona del siguiente modo: Cuando el cliente sin disco arranca, envía un mensaje broadcast de tipo RARP, preguntando si alguien le puede proporcionar si dirección IP. Si alguna de las máquinas presentes en la red conoce la dirección del cliente, le devuelve un mensaje de respuesta RARP informándole de su dirección IP. El cliente puede recibir múltiples respuestas RARP. Es este caso se queda con la primera respuesta que recibe.

2.1.7. Organización de redes en subredes(subneting)

La mayoría de las redes de tamaño grande (clase A o B) se dividen en subredes, aunque también las redes pequeñas (clase C) de pueden dividir en subredes. Normalmente, las distintas subredes se suelen unir entre sí mediante switches, puentes o routers. Las ventajas de dividir una red en varias subredes son las siguientes:

- Permite aislar el tráfico entre las distintas subredes, con lo cual se reduce el trafico global.
- Permite limitar y proteger el acceso a las distintas subredes.
- Permite organizar la red en áreas o departamentos, de manera que se asigna a cada departamento un subconjunto de direcciones IP. La gestión de las direcciones IP asignadas a un departamento, se puede

delegar en el propio departamento, con lo cual se descentraliza la tarea de asignación de direcciones y se facilita la tarea del administrador de la red.

Por ejemplo, una organización con una red de clase B con dirección 150.20.0.0 podría dividir esta red en 256 subredes de la siguiente forma:

- Subred 0: 150.20.0.0
- Subred 1: 150.20.1.0
- Subred 2: 150.20.2.0
- Subred 255: 150.20.255.0

Los dos primeros bytes identifican a la red (fijados por NIC), el tercer byte identifica la subred y el último byte identifica a la estación dentro de la subred (150.20.subred.estacion).

Cuando una red se divide e varias subredes, se debe utilizar una máscara de subred diferente a la máscara global de la red. En el ejemplo anterior la máscara de subred debe valer 255.255.255.0.

Nótese que el número de estaciones por subred es de 254, pues la dirección terminada en 0 identifica a la subred y la dirección terminada en 255 es la dirección broadcast de la subred.

En este ejemplo se ha dividido una red de clase B en 256 subredes con 254 estaciones cada una. Sin embargo, en el caso mas general, se podría dividir en un mayor número de subredes con menos estaciones por subred o al contrario, un menor número se subredes con más estaciones por subred. Aplicando esta misma idea, también resulta posible dividir redes de la clase C en varias subredes.

2.2. IPv6

2.2.1. Introducción

IPv6 es la versión 6 del protocolo de red IP. Fue diseñado por Steve Deering y Craig Mudge para sustituir a IPv4 debido a sus numerosas deficiencias. Fue adoptado por el IETF (Internet Engineering Task Force) en 1994 y es la segunda versión del protocolo IP que se adopta para su uso general.

2.2.2. Diferencias con IPv4

Existen numerosas diferencias entre IPv4 e IPv6. Precisamente, son estas diferencias las que motivan la implantación de IPv6.

2.2.2.1. Mayor espacio de direcciones

Las direcciones en IPv6 tienen 128 bits de longitud, frente a los 32 bits de las direcciones de IPv4. Esto supone, teóricamente, unas 3.4x10^38 direcciones IPv6 frente a las 4294967296 direcciones que permite IPv4.

La ventaja principal de este incremento en el espacio de direcciones se refleja, sobre todo, en que se elimina la limitación actual de disponibilidad de direcciones IP y los métodos artificiales para evitar dicha limitación como, por ejemplo, NAT (Network address translation). La principal desventaja de este aumento del número de direcciones es la sobrecarga en el ancho de banda utilizado debido al incremento del tamaño de la cabecera de los paquetes. Esta sobrecarga se puede aliviar utilizando la opción de compresión de las cabeceras de los mensajes.

2.2.2.2. Autoconfiguración

Los hosts pueden configurarse automáticamente al conectarse a una red IPv6. El nuevo host manda una petición a los routers existentes en la red para que le indiquen los parámetros de configuración que tiene que utilizar. Si los routers están preparados para ello, contestan con la información solicitada y el host se configura acorde a los parámetros recibidos.

Si los routers no están preparados para proporcionar este tipo de parámetros,los nodos pueden utilizar el protocolo DHCPv6, que constituye el equivalente al DHCP de IPv4, para autoconfigurarse o pueden configurarse manualmente.

Los host son los únicos que pueden beneficiarse de la autoconfiguración, ya que los routers deben ser configurados manualmente.

2.2.2.3. Jumbogramas

El tamaño de los paquetes en IPv4 está limitado a 64 KB por paquete, mientras que IPv6 permite trabajar con paquetes de hasta 4 GB llamados jumbogramas. Esto permite mejorar el rendimiento en redes con un MTU (Maximum Transmission Unit) alto.

Debido a las limitaciones de TCP y UDP respecto al tamaño de los paquetes, se deben realizar algunos ajustes en la capa de transporte si se quieren utilizar jumbogramas.

2.2.2.4. IPSEC

IPSEC es un protocolo de encriptación y autenticación a nivel de capa de red. Su soporte en IPv6 es obligatorio, mientras que en IPv4 es opcional. Debido a la falta de seguridad en IPv4, con IPsec se prentende proporcionar: integridad y encriptación de las comunicaciones y autenticación de los extremos de la comunicación.

2.2.2.5. IP Multicast

IP Multicast es un método para enviar datagramas a varios receptores a la vez (comunicación uno a muchos). Su soporte es obligatorio en IPv6, al contrario que en IPv4, donde es opcional. Esta técnica se puede utilizar a varios niveles: a nivel de red (igual que un paquete multicast de IPv4), a nivel de organización (para todo un conjunto de direcciones IPv6 correspondientes a una misma entidad de gestión) o a nivel global (como intenta conseguir el proyecto m6bone).

2.2.2.6. Mobilidad

Mobile IPv6 es una característica de IPv6 que permite a un nodo cambiar de ubicación y seguir manteniendo sus conexiones actuales. Esta característica es de obligatorio soporte en IPv6, mientras que es opcional en IPv4.

2.2.3. Direccionamiento

Como ya se comentó anteriormente, las direcciones IPv6 tienen 128 bits de longitud. Para representarlas, se utilizan 8 grupos de 4 dígitos hexadecimales cada uno, separados por el símbolos de dos puntos.

```
2001:0db8:85a3:08d3:0000:0000:0000:0001
```

Existen varios convenios para simplificar las direcciones IPv6 y facilitar su uso:

1. La cadena más larga de ceros se puede sustituir por dos símbolos de dos puntos. Sólo se puede utilizar una vez los dos símbolos de dos puntos dentro de una dirección IPv6.

   Table 2-1. Resumiendo direcciones IPv6 (eliminando ceros consecutivos)

Antes del cambio	Después del cambio
2001:0db8:85a3:08d3:0000:0000:0102:0304	2001:0db8:85a3:08d3::0102:0304

2. En cada uno de los 8 grupos de la dirección, se pueden omitir los ceros a la izquierda.

Table 2-2. Resumiendo direcciones IPv6 (eliminando ceros a la izquierda)

Antes del cambio	Después del cambio
2001:0db8:85a3:08d3::0102:0304	2001:db8:85a3:8d3::102:304

3. Los últimos 4 bytes de la dirección (los que están más a la derecha de la misma) pueden escribirse en notación decimal usando puntos como separadores. Esto se mantiene por razones de compatibilidad a la hora de utilizar direcciones IPv4 dentro de las direcciones IPv6.

Table 2-3. Compatibilidad direcciones IPv4 en IPv6

Antes del cambio	Después del cambio
2001:db8:85a3:8d3::102:304	2001:db8:85a3:8d3::1.2.3.4

Cuando se utilizan como URLs, las direcciones IPv6 van entre corchetes. Si se quiere especificar el puerto al que conectarse, se deja fuera del corchete para no crear confusión con el último campo de la dirección.

```
http://[2001:db8:85a3:8d3::1]
http://[2001:db8:85a3:8d3::1]:80
```

Al igual que en IPv4, las redes en IPv6 se definen utilizando la notación CIDR. Esto es, se escribe el prefijo de red y después se especifica la longitud de la red en notación decimal.

Por ejemplo, la red 2001:db8:85a3:8d3::/64 comprende los hosts desde 2001:db8:85a3:8d3:: hasta 2001:db8:85a3:8d3:ffff:ffff:ffff:ffff .

Los tipos de direcciones IPv6 se dividen en 3 categorías:

- Unicast: definen un único interfaz. Los paquetes enviados a una dirección unicast llegan únicamente a dicha dirección.
- Anycast: se asignan a interfaces distintos de varios nodos. Los paquetes enviados a una dirección anycast llegan sólo al interfaz del nodo más cercano al emisor, según el protocolo de enrutado.
- Multicast: se asignan a varios interfaces de uno o varios nodos. Los paquetes enviados a una dirección multicast llegan a todos los interfaces que tengan dicha dirección.

En IPv6, al igual que en IPv4, hay direcciones con significados especiales:

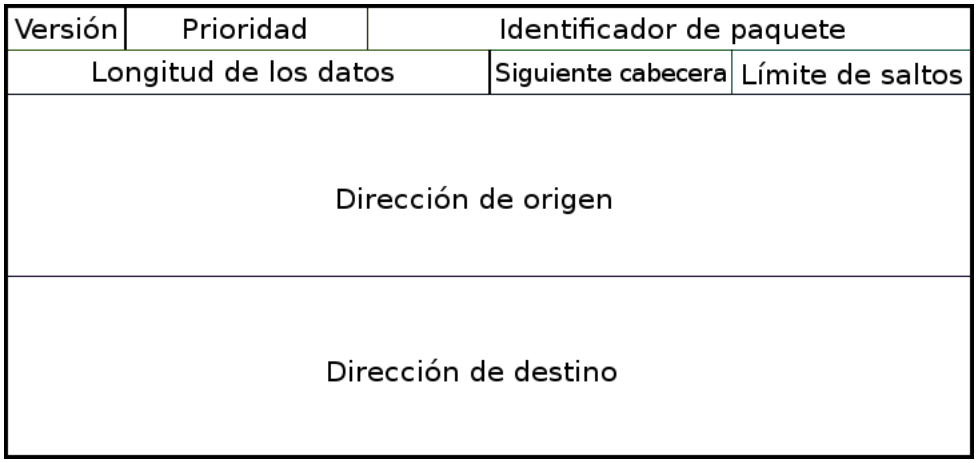
Table 2-4. Significado de las direcciones en IPv6

Dirección	Significado
::/128	Reservada y no enrutable
::1/128	Dirección loopback (equivale a 127.0.0.1 en IPv4)
::ffff:0:0/96	Direcciones IPv4 mapeadas en IPv6
fc00::/7	Estas direcciones sólo se pueden enrutar entre unos nodos predeterminados de antemano
fe80::/64	Estas direcciones se utilizan en los mecanismos de autoconfiguración y no pueden salir de la red local
ff00::/8	Direcciones multicast

2.2.4. Formato de un paquete IPv6

Los paquetes IPv6 están compuestos por dos partes: la cabecera y la información transmitida.

Figure 2-2. Esquema de un paquete IPv6



La cabecera de un paquete IPv6 contiene los siguientes campos:

- Versión del protocolo IP (4 bits)
- Prioridad (1 byte)
- Identificador (20 bits)
- Longitud de los datos contenidos en el paquete (2 bytes).
- Siguiente cabecera (1 byte)

- Límite de saltos que puede dar el paquete (1 byte)
- Dirección de origen (128 bits)
- Dirección de destino (128 bits)

El tamaño normal de los datos de un paquete es de 64KB, pero se puede aumentar con la opción "jumbo payload". La fragmentación de los paquetes se realiza en el emisor, nunca en los enrutadores, al contrario de lo que ocurría en IPv4. El emisor se encarga de ajustar el tamaño de los paquetes a los MTUs correspondiente mediante el descubrimiento de dichos MTUs.

Utilizando el campo de "Siguiente cabecera" se pueden especificar opciones adicionales de configuración para el paquete. Cada cabecera adicional debe tener una longitud múltiplos de 8 octetos para que siempre puedan ser alineadas. Deben leerse todas las cabeceras en el orden indicado por el paquete. Dichas cabeceras sólo son examinadas y procesadas por el nodo de destino, salvo que éste sea una dirección multicast. En ese caso, todos los nodos pertenecientes a ese grupo determinado de multicast leen y procesan las todas las cabeceras adicionales.

Las cabeceras adicionales disponibles son las siguientes y se deben especificar en este orden:

Table 2-5. Cabeceras adicionales de un paquete IPv6

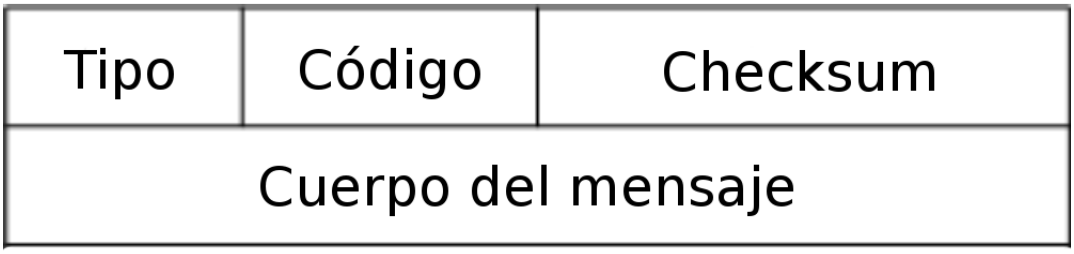
Cabecera	Propósito
Opciones de salto a salto	Incluye información opcional que debe ser examinada por todos y cada uno de los nodos por los que pase el paquete.Son la excepción, ya que, como se comentó anteriormente, las cabeceras adicionales son procesadas únicamente por el host de destino.
Opciones de enrutado	Se utiliza para dar una lista de nodos por los que debe pasar el paquete de camino a su destino.
Opciones de fragmentación	Se utiliza para indicarle al host de destino que se trata de un fragmento de paquete y no del paquete entero.
Opciones de autenticación	Sirven para determinar la integridad y autenticidad de un paquete.
Opciones de encapsulado seguro de los datos	Se utilizan para conseguir confidencialidad en los datos enviados dentro del paquete.
Opciones de destino	Estas opciones son analizadas únicamente por el host de destino.

2.2.5. ICMPv6

ICMPv6 (Internet Control Message Protocol Version 6) es la versión de ICMP para IPv6. A diferencia de IPv4, en el que los firewalls podían filtrar todo el tráfico ICMP sin perder conectividad, en IPv6 es parte integral del protocolo y todo los nodos deben soportarlo obligatoriamente. En esta nueva versión, combina las funcionalidades del anterior ICMP, la de IGMP (Internet Group Membership Protocol) y la de ARP (Address Resolution Protocol). También se ha aprovechado para simplificar el protocolo y eliminar algunos tipo obsoletos de mensajes que estaban en desuso.

Los mensajes ICMPv6 se envían dentro de paquetes IPv6 cuyo valor del campo de identificador de la siguiente cabecera es 58. Las funciones de este protocolo son informar de los errores encontrados, realizar diagnósticos, realizar el descubrimiento de nodos vecinos (Neighbor Discovery).

Figure 2-3. Esquema de un paquete ICMPv6



Los campos de un mensaje ICMPv6 son los siguientes:

- Tipo: Indica el tipo de mensaje (1 byte).
- Código: Depende del tipo del mensaje y añade información adicional (1 byte).
- Checksum: Se usa para comprobar que el mensaje no ha llegado corrupto (2 bytes).
- Cuerpo del mensaje: Depende del tipo y código del mensaje (longitud variable).

Existen dos tipos de mensajes en ICMPv6: mensajes de error y mensajes de información.

2.2.5.1. Mensajes de error

Existen cuatro tipos de mensajes de error:

Table 2-6. Códigos de los mensajes de error de ICMPv6

Número de mensaje	Tipo	Significado

Número de mensaje	Tipo	Significado
1	Destino inalcanzable	No se pudo entregar un paquete.
2	Paquete demasiado grande	Un paquete no se pudo reenviar por parte de un router porque su tamaño era mayor que el MTU de ese enlace.
3	Tiempo excedido	Se sobrepasó el número máximo de saltos de un paquete.
4	Problema de parámetro	No se pudo procesar un paquete debido a que alguno de sus campos (o el de alguna de sus extensiones) no pudo ser identificado.

2.2.5.2. Mensajes de información

Existen varios tipos de mensajes de error. Los más importantes son:

Table 2-7. Tipos de mensajes de error en ICMPv6

Número de mensaje	Tipo	Significado
128	Petición de eco	Equivale a una petición PING en ICMPv4.
129	Respuesta de eco	Equivale a una respuesta a un PING en ICMPv4
133	Solicitud de router	Un nodo nuevo pide información sobre los routers de la red para autoconfigurarse.
134	Anuncio de router	Los routers de la red se anuncian en la misma para que los demás nodos lo sepan y se puedan autoconfigurar.
135	Solicitud de vecino	Equivalente a una petición ARP en IPv4.
136	Anuncio de vecino	Equivalente a una respuesta ARP en IPv4.

2.2.5.3. Descubrimiento de vecinos

En IPv4 se utiliza ARP (Address Resolution Protocol) para determinar la dirección MAC de los hosts de

una misma red. En IPv6, en cambio, esta funcionalidad ha sido incorporada a ICMPv6 y aumentada para proporcionar más prestaciones mediante el mecanismo de descubrimiento de vecinos, con lo que se elimina la necesidad de usar un protocolo fuera de IP, como es el caso de ARP.

Las funciones que realiza el descubrimiento de vecinos son las siguientes:

- Determinar las direcciones MAC de los hosts de una misma red.
- Encontrar routers vecinos que puedan reenviar paquetes fuera de la red.
- Llevar un control sobre qué vecinos son alcanzables y cuáles no para detectar cambios en las direcciones MAC.

Todas estas funciones se llevan a cabo mediante los mensajes de información ICMPv6 de solicitud de router, anuncio de router, solicitud de vecino y anuncio de vecino.

2.2.5.4. Descubrimiento del MTU de las rutas

En IPv4, los routers son los encargados de fragmentar un paquete si al entrar a una red el MTU (Maximun Transfer Unit) de dicha red es menor que el tamaño del paquete IPv4. Los fragmentos producidos se pueden volver a fragmentar al llegar a otra red si se produce una situación similar. Finalmente, todos los fragmentos son reensamblados en el host destino para reconstriur el paquete IPv4 original.

Esta situación cambia completamente en IPv6, ya que en este caso es el nodo emisor el encargado de averiguar cuál es el mínimo MTU de entre todos los que MTUs de las redes por las que va a pasar el paquete. Esta técnica libera a los routers de la sobrecarga que les producía realizar esta operación.

El proceso para descubrir el mínimo MTU de toda la ruta es el siguiente:

1. Se manda el paquete con un tamaño inicial igual al del MTU de la red del host emisor.
2. Si alguno de los routers en la ruta del paquete detecta que el tamaño del paquete es mayor que el MTu de su red, le manda un mensaje ICMPv6 de "paquete demasiado grande" y le indica el MTU de esa red.
3. El host emisor reduce el tamaño del paquete al MTU recibido en el mensaje ICMPv6 y vuelve a mandar el paquete.
4. La operación se repite hasta que el paquete llega a su destino o se alcance el MTU mínimo que 1280 bytes, momento en el que no se sigue reduciendo el tamaño del paquete.

El MTU de una ruta puede cambiar a lo largo del tiempo, por lo que los hosts emisores tratan de incrementar el tamaño de sus mensajes cada cierto tiempo para comprobar si se ha producido dicho

cambio. En el caso de que el destino sea una dirección multicast, el tamaño del mensaje tendrá que ajustarse al del menor MTU existente en las redes de los hosts pertenecientes a ese grupo multicast.

2.2.5.5. Autoconfiguración de los hosts

En IPv4, si se desea que los hosts se autoconfiguren, hay que utilizar el protocolo DHCP (Dynamic Host Configuration Protocol) perteneciente a la capa de aplicación. Mediante este protocolo, los hosts le preguntan a los routers de la red que estén escuchando peticiones DHCP qué IP pueden asignarse y qué opciones de configuración deben utilizar.

En IPv6, esta característica esta incluida en la propia capa de red (autoconfiguración "stateless") y no se necesita de software adicional para proporcionarla, aunque también se puede utilizar DHCPv6, el equivalente a DHCP pero para IPv6, si se quiere mantener el anterior modelo de autoconfiguración (autoconfiguración "stateful"). Ambos tipos de autoconfiguración pueden convivir perfectamente en la misma red. Es frecuente usar la autoconfiguración "stateless" para conseguir una IPv6 y después usar la autoconfiguración "stateful" para conseguir otros parámetros de configuración, como por ejemplo, los servidores DNS disponibles en esa red.

El hecho de que el mecanismo de autoconfiguración esté incluido en el propio protocolo IPv6 facilita mucho la adopción de esta tecnología por parte de dispositivos que no sean ordenadores personales o servidores, como pueden ser teléfonos móviles, electrodomésticos, consolas de videojuegos, etc. Esta característica les permite autoconfigurarse en cuanto estén en una red, ya sea cableada o inalámbrica, sin necesidad de ninguna manipulación por parte de los usuarios.

En el caso de los routers, no se puedem utilizar estos mecanismos de autoconfiguración, ya que los routers se deben configurar manualmente y de forma estática, pera que puedan servir de referencia a los demás hosts de la red a la hora de autoconfigurarse.

En el proceso de autoconfiguración "stateless" se siguen los siguientes pasos:

1. Se genera una dirección de enlace local (link-local) con prefijo FE80 y se le añade el identificador de interfaz. Esta dirección es una primera tentativa de autoasignarse una dirección única dentro de la red.
2. El host se añade al grupo multicast de todos los nodos (FF02::1) y al grupo multicast correspondiente a la dirección de enlace local que se asignó previamente.
3. Se envía un mensaje ICMPv6 de solicitud de vecino al grupo multicast correspondiente a la dirección de enlace local que se generó en el primer caso. Si se recibe un mensaje ICMPv6 de aviso de vecino quiere decir que dicha dirección ya estaba ocupada y este host no se puede autoconfigurar automáticamente, por lo que requerirá una intervención manual por parte del propio usuario. Si no se recibe ningún mensaje de este tipo, el host se asigna la dirección de enlace local generada al comienzo del proceso.
4. Una vez el host tiene una dirección única dentro de la red, manda un mensaje ICMPv6 de solicitud de router al grupo multicast de todos los routers de la red (FF02::2) para saber cuántos routers hay

en la red y que prefijos de red anuncia cada uno.

5. Todos los routers de la red contestan con un mensaje ICMPv6 de anuncio de router con el prefijo que anuncia cada uno. El host coge dichos prefijos y los combina con su identificador de interfaz para generar una dirección IPv6 global por cada uno de los routers. Todas estas direcciones IPv6 se asignan al mismo interfaz de red, que puede tener tantas direcciones IPv6 como desee.

2.2.6. Protocolos de enrutado

Protocolos de enrutado para IPv6 y una breve descripción.

2.2.6.1. RIP

RIP (Routing Information Protocol) es un protocolo de enrutado interior basado en algoritmos de distancia-vector. Está diseñado para ayudar con el enrutado interno de una red, informando de las redes con las que comunica cada router y a qué distancia estan dichas redes. Aunque está considerado como un protocolo obsoleto, todavía se sigue utilizando en muchas redes.

RIPng es una extensión de RIP para redes que utilicen IPv6. Notifica de cambios en la red mediante mensajes multicast a la dirección ff02::9 .

2.2.6.2. OSPF

El protocolo OSPF (Open Shortest Path First) es otro protocolo de enrutado interior que, a diferencia de RIP, utiliza el algoritmo de Dijkstra para calcular las rutas de enrutado más cortas. OSPF es el protocolo de enrutado interior más usado a nivel global y utiliza el protocolo IP directamente para mandar sus mensajes, sin usar TCP ni UDP como hace RIP.

OSPFv3 es la nueva versión de OSPF que soporta redes IPv6.

2.2.6.3. BGP

Al contrario que RIP y OSPF, BGP (Border Gateway Protocol) es un protocolo de enrutado externo. Maneja una lista de redes alcanzables entre sistemas autónomos para conseguir un enrutado descentralizado.

Existen unas extensiones para que BGp que le permiten realizar enrutados entre redes IPv6.

2.2.7. Implantación en una red

Hasta que se alcance un punto en el que IPv6 suplante a IPv4 y se deje de utilizar este último, hecho que no parece probable en un futuro cercano, se intenta hacer que ambas tecnologías convivan en las distintas redes existentes. Existen varios métodos para conseguir este fin: configuración "Dual stack" o túneles.

2.2.7.1. Configuración "Dual stack"

Esta opción consiste en que las máquinas de una red soporten tanto IPv4 como IPv6. También se permite que haya máquinas que sólo entiendan IPv4, pero se intenta que utilicen IPv6 el máximo número posible de máquinas. Esta opción de configuración es la más común y la más comprometida con IPv6.

2.2.7.2. Túneles IPv6

Esa técnica se utiliza cuando una máquina de una red IPv4 quiere comunicarse con otra de una red IPv6. Para ello, se encapsulan los paquetes IPv6 en paquetes IPv4.

Existen dos tipos de túneles:

- Túneles manuales: los extremos del túnel se configuran explícitamente, normalmente mediante "Tunnel brokers". Esta opción es la elegida por las grandes redes por motivos de facilidad de administración y mantenimiento.
- Túneles "6to4": este tipo de túneles no requieren una configuración manual. Son utilizados en la fase inicial de transición a un sistema IPv6 nativo y están pensados como un mecanismo temporal para permitir la conectividad, no como un sistema permanente y estable.

2.2.8. Estado actual de IPv6

El gobierno de los Estados Unidos ha ordenado el despliegue de IPv6 por todas sus agencias federales para el verano de 2008. Para ello invertirá 5,000 millones de dólares, más del doble de lo que se gastó para evitar el efecto 2000.

China planea que en las Olimpiadas de Beijing 2008 todo esté montado sobre IPv6, desde cámaras de seguridad, taxis e incluso las cámaras que grabarán los Juegos Olímpicos y luego los retransmitirán por streaming por internet.

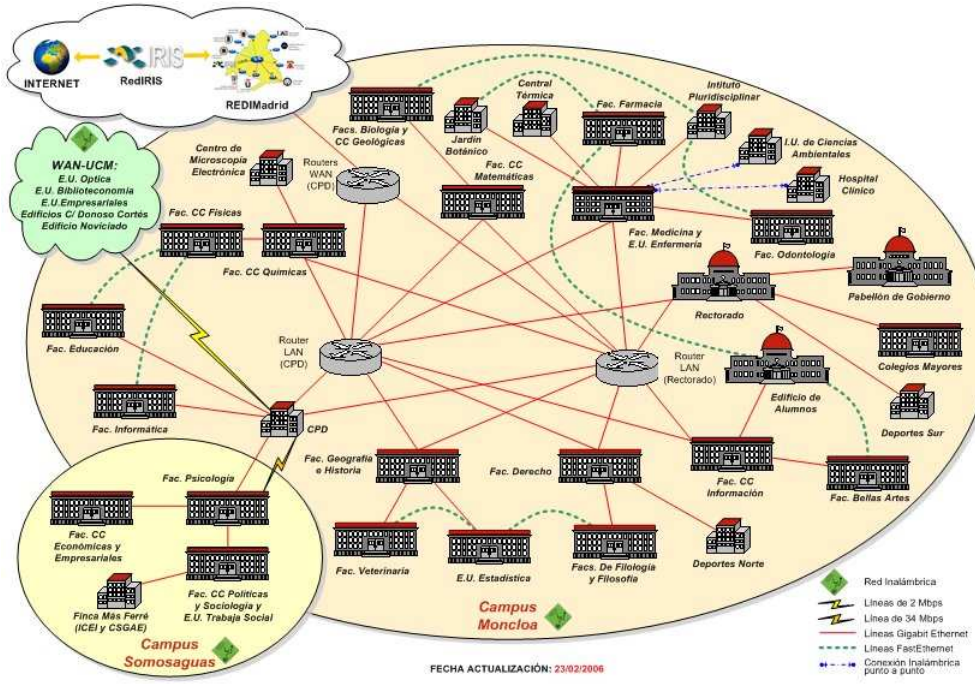
Se espera que IPv4 se siga soportando hasta por lo menos el 2025, dado que hay muchos dispositivos que no se migrarán nunca a IPv6 y que seguirán siendo utilizados por mucho tiempo.

Chapter 3. Caso de estudio Red de Datos de la UCM

3.1. Situación actual de la red de datos de la UCM

La red de datos de la UCM interconecta y da acceso a internet a todos los centros de la Complutense, así como a otros centros relacionados con ella. Consta de dos grandes routers centrales, uno en el centro de procesamiento de cálculo (CPD) y otro en el edificio de vicerrectorado, formando una configuración física en forma de doble estrella. El segundo router sirve para dotar a la red de tolerancia a fallos y prevenir de averías y saturación de tráfico en el primer router.

Figure 3-1. Esquema de la red de datos de la UCM



Las conexiones que unen los edificios y los routers centrales son de tipo "Gigabit Ethernet". No todos los centros enlazan directamente con los routers centrales. Algunos centros enlazan con otros edificios que hacen de intermediario entre ellos y los routers centrales. El enlace con el campus de Somosaguas también es del tipo Gigabit Ethernet y además existe otro enlace a 34 Mbps como respaldo ante posibles fallos en el enlace principal. Hay conexiones de tipo "Fast Ethernet" entre algunos de los centros, pero actualmente se están cambiando para conseguir que todas las conexiones LAN físicas entre los centros

sean, al menos, del tipo "Gigabit Ethernet".

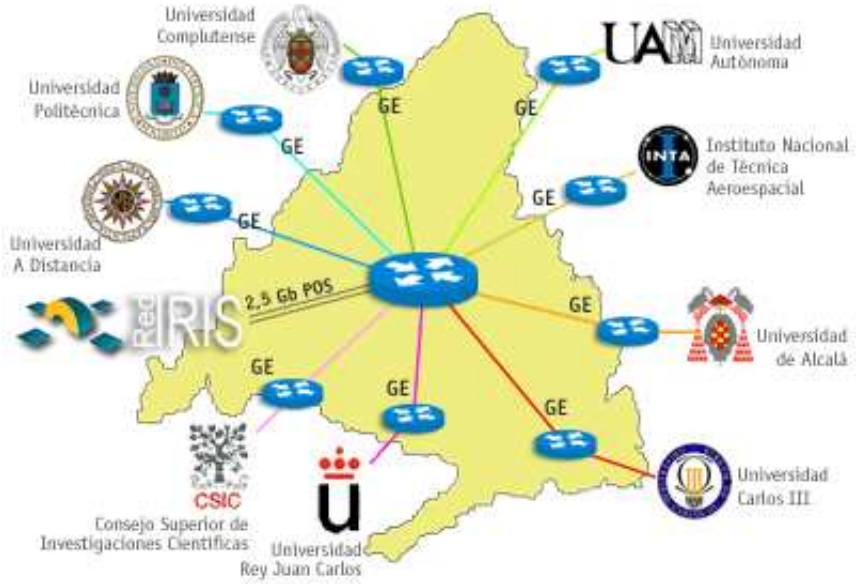
Existen enlaces de tipo WAN a 2 Mbps con los siguientes centros:

- E.U. EE. Empresariales.
- E.U. de Biblioteconomía y Documentación.
- I.C.E .
- E.U. de Óptica.
- Edificios de la calle Donoso Cortés.
- Edificio de Noviciado: Biblioteca Marqués de Valdecilla y E. de Relaciones Laborales.

También existen conexiones inalámbricas punto a punto con el I.U de ciencias ambientales y con el Hospital Clínico.

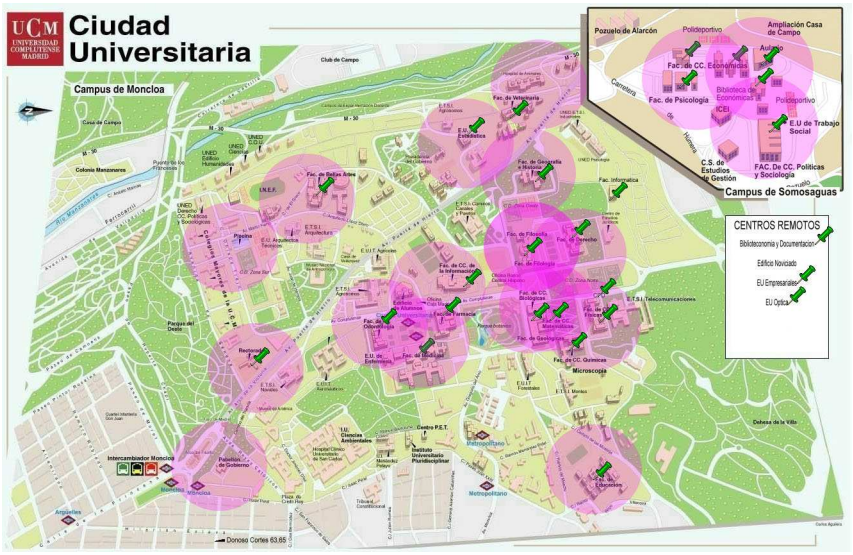
La conexión a internet se realiza mediante un enlace del tipo "Gigabit Ehternet" con la Red Telemática de Investigación de Madrid (REDIMadrid), al igual que lo hacen el resto de las universidades públicas de la Comunidad de Madrid. REDIMadrid se conecta a internet a través de la Red Académica Nacional (RedIRIS) mediante un enlace de 2,5Gb .

Figure 3-2. Conexiones externas de la UCM



La red inalámbrica de la UCM soporta tanto el estándar 802.11b como el 802.11g a través de puntos de acceso certificados como Wi-Fi.

Figure 3-3. Plano de conectividad inalámbrica en la red de datos de la UCM



La red de datos también permite conexiones remotas. Este tipos de conexiones se pueden realizar vía modem, utilizando la red telefónica básica (RTB) o una conexión RDSI, o mediante el uso de una VPN (Virtual Private Network).

En cuanto a los puestos de trabajo para los usuarios, se proporciona a cada puesto una capacidad de acceso a la red mínima de 10 Mbps y, en la mayoría de los casos, de 100 Mbps, siempre conmutados. Dentro de la red de datos de la UCM hay un total de 40.764 puntos de acceso instalados repartidos de la siguiente manera: Campus de Moncloa 31.435, Campus de Somosaguas 7.275 y Centros Remotos 2.054. (Datos de marzo de 2006)

3.1.1. Características físicas de la red

Toda la red se basa en elementos modulares, compatibles e intercambiables entre sí, lo cual permite una gran facilidad de adaptación a los cambios que requieran ampliación o actualización de las conexiones. Por otro lado, la electrónica cuenta con elementos redundantes, como las fuentes de alimentación, con capacidad de reparto de la carga, y todos los módulos son reemplazables "en caliente", sin interrupción del servicio. La capacidad de los mecanismos de conmutación es suficiente para soportar el uso de al menos el 80% del ancho de banda agregado del máximo número de puntos instalables en cada módulo y chasis, sin congestión.

Los chasis modulares aceptan la inclusión de módulos que soportan los siguientes protocolos y estándares:

- Ethernet, Fast Ethernet, Gigabit Ethernet, FDDI.
- Gestión SNMP y RMON.
- Redes virtuales (802.1 Q).
- Calidad de servicio (802.1 p).
- Control de flujo (802.3 x).
- Soporte multicast.

Además, el equipamiento ofrece las siguientes funcionalidades relevantes:

- Agregación de líneas o port trunking.
- Port mirroring, que permita redirigir el tráfico en una puerta a otra puerta para ser observado.
- Establecimiento de filtros de acceso sobre la información de niveles 2, 3 y 4.
- Establecimiento de políticas de seguridad asociadas a cada puerto.
- Control de los umbrales de tráfico de broadcast.
- Limitación del uso del ancho de banda por puerto.

3.1.2. La doble estrella física

Todos los hosts que pertenecen a la red de datos de la UCM están conectados a un único router central (instalado en el edificio del CPD) que se encarga de darles conectividad con Internet. Sin embargo, por razones de tolerancia a fallos hay un segundo router de apoyo (instalado en el edificio del Vicerrectorado). El objetivo que se persigue con esta redundancia es que si el router primario deja de funcionar, todo el tráfico pasaría por el router secundario y se seguiría manteniendo la conectividad con Internet en toda la UCM. Para conseguir este comportamiento se utiliza VRRP (Virtual Router Redundancy Protocol).

VRRP es un protocolo de redundancia diseñado para aumentar la tolerancia a fallos de la puerta de enlace predeterminada de la red. Este aumento en la fiabilidad de la red se consigue mediante la utiliazación de un "router virtual" como puerta de enlace predeterminada para la subred, dicho "router virtual" actúa como representante de un grupo de routers formado por un master y uno o varios routers de reespaldo. De este modo, en lugar de tener un único router para la red existe un grupo de routers con un representante. La elección de este representante es dinámica y está sujeta a la disponibilidad de los routers que han sido asignados para formar parte del grupo de routers.

Los routers físicos de la red se comunican entre ellos utilizando mensajes multicast a la IP 224.0.0.18 con el número de protocolo 112. Si transcurrido el período de tiempo indicado por el administrador no se

recibe el mensaje del master, uno de los routers de reespaldo toma el papel de master y se encarga de las funciones de encaminamiento. El proceso es transparente para los usuarios.

3.2. Particionado lógico de la red

Como se ha explicado anteriormente, la red de datos de la UCM es una gran LAN (Red de Área Local), la tecnología que subyace en la mayoría de la red es Ethernet o alguna de sus mejoras Fast Ethernet o Gigabit Ethernet. El protocolo Ethernet se encarga de controlar el acceso de cada host a un medio compartido (cable de red). El problema de utilizar la tecnología Ethernet es que disminuye su eficiencia si se introducen muchos hosts en el mismo segmento de red. Por ello, al tener una red tan grande es necesario realizar algún tipo de particionado lógico de la red, es decir, partir una red grande en varias subredes para mejorar tanto la eficiencia en las comunicaciones como la gestión de la propia red.

La Universidad Complutense de Madrid recibe de RedIris la siguiente subred IPv4 de clase B: 147.96.0.0/16. Esta red es de acceso público, es decir, un usuario desde su casa puede leer la página web de la UCM o de cualquiera de las facultades y el servidor Web que hospeda dicha página estará dentro de esta subred de clase B.

3.2.1. Particionado en subredes

Siguiendo la especificación de IP dentro de la ucm se pueden numerar equipos desde la dirección 147.96.1.1 hasta 147.96.255.255, esto permite tener hasta 65000 equipos dentro de esta subred. La administración de una única subred de 65000 es muy complicada por lo que es necesario dividir esta subred de clase B en múltiples subredes de clase C. Las redes IPv4 de clase C son de la siguiente manera A.B.C.D/32, por lo que la distribucion de subredes en la red de datos de la UCM quedaría de la suiguiente manera: Hay 255 subredes de clase C, van desde la subred 147.96.1.0/24 hasta la subred 147.96.255.0/24.

Gracias a esta división en subredes, se pueden distribuir subredes a los distintos edificios en función de la demanda de puestos de cada uno. Es necesario reseñar que en la UCM no sólo hay facultades sino que también hay edificios como el Centro de Proceso de Datos o el Vicerrectorado cuya misión no es la docencia sino la administración y también necesitan estar interconectados a la misma red.

3.2.2. Particionado en VLANs

Una VLAN (Virtaul LAN o Red de Área Local Virtual) es un método para crear redes lógicas independientes detro de una red física. Varias VLANs pueden coexistir dentro de dicha red física. Éstos ayuda a reducir el dominio de broadcast y ayuda a la administración separando segmentos lógicos de una LAN pero sin modificar los datos que transcurren por dicha red (se sigue manteniendo el mismo enrutado). Una VLAN consiste en una red de computadores que se comporta como si estuviera

conectada al mismo cable (incluso aunque realmente las computadoras se encuentren físicamente conectadas a distintos segmentos de la LAN).

Los administradores de la red pueden configurar las VLANs mediante el uso de hardware específico o mediante software, lo cual hace a este sistema extremadamente flexible. Una de las mayores ventajas del uso de VLANs es que se puede mover físicamente un ordenador de un sitio a otro sin que sea necesario ninguna reconfiguración de hardware.

Las ventajas que reporta el uso de VLANs dentro de la red de datos de la UCM es el siguiente:

- Incrementa el número de dominios de broadcast pero reduce el tamaño de dichos dominios. Esto se traduce en una reducción del tráfico de la red y un incremento de la seguridad (ambos objetivos se ven muy perjudicados en redes muy grandes)
- Reducción del trabajo de mantenimiento para crear nuevas subredes.
- Reducción de los requerimientos hardware dado que las redes pueden estar separadas de forma lógica en lugar de forma física.
- Aumento en el control a través de muchos tipos de tráfico.
- Creación de múltiples switches lógicos dentro de un swithc físico.

3.3. Aplicaciones y servidores

Este capítulo contendrá un análisis de las aplicaciones y servidores utilizados en la UCM.

3.3.1. Servidores Web

El servidor Web más importante de la Universidad Complutense de Madrid es goliat.sim.ucm.es cuya IP es 147.96.1.15, este servidor contiene la página Web http://www.ucm.es y las siguientes facultades tienen una página web dependiente de este servidor:

- Facultad de Bellas Artes
- Facultad de Ciencias Biológicas
- Facultad de Ciencias de la Documentación
- Facultad de Ciencias de la Información
- Facultad de Ciencias Económicas y Empresariales
- Facultad de Ciencias Físicas
- Facultad de Ciencias Geológicas
- Facultad de Ciencias Políticas y Sociología

- Facultad de Ciencias Químicas
- Facultad de Derecho
- Facultad de Farmacia
- Facultad de Geografía e Historia
- Facultad de Medicina
- Facultad de Odontología
- Facultad de Psicología
- Facultad de Veterinaria

Además del servidor goliat.sim.ucm.es hay otros servidores Web que se encargan de almacenar las páginas Web de algunas facultades. Dichos servidores se listan a continuación:

- Facultad de Educación: sergeredu.ccedu.ucm.es (147.96.43.9). Página Web: http://www.edu.ucm.es
- Facultad de Informática: 147.96.85.71. Página Web: http://www.fdi.ucm.es
- Facultad de Filología: filoserver.filol.ucm.es (147.96.41.200). Página Web: http://www.filol.ucm.es
- Facultad de Filosofía: fs-morente.filos.ucm.es (147.96.40.129). Página Web: http://fs-morente.filos.ucm.es
- Facultad de Matemáticas: matnfs.mat.ucm.es (147.96.20.227). Página Web: http://www.mat.ucm.es

3.3.2. Servidores DNS

La red de datos de la UCM tiene dos servidores de nombres (servidores DNS). El servidor DNS primario es ucdns.sis.ucm.es (147.96.2.4) y el servidor DNS secundario es crispin.sim.ucm.es (147.96.1.9). Existen varios motivos para la existencia de dos servidores DNS en lugar de uno. El primero de ellos es por tolerancia a fallos. Si el servidor primario dejase de funcionar, todas las peticiones de resolución de nombres se derivarían al servidor secundario y el usuario final no notaría la diferencia. Otro motivo no menos importante es que teniendo varios servidores DNS se puede distribuir la carga de proceso entre varios equipos evitando la sobrecarga de un único servidor.

Es interesante recordar que si existen varios servidores DNS dentro de una misma organización (en este caso la Universidad Complutense de Madrid) no es necesario que todos los servidores dispongan de la base de datos completa con la correspondencias de nombres e IPs. Gracias al funcionamiento del protocolo DNS, si un servidor no contiene en su base de datos local el valor que debe devolver al usuario, se encargará de redirigir la petición a otro servidor hasta dar con el valor solicitado.

3.3.3. Servidor de Correo

La Universidad Complutense de Madrid ofrece servicio a los usuarios de correo electrónico ubicado en UCIMAP, UCPOP y PASMAIL.PAS, del dominio UCM.ES.

Table 3-1. Datos del servidor de correo

Servidor	Nombre	IP
Servidor POP	ucpop.ucm.es	147.96.1.161
Servidor IMAP	ucimap.ucm.es	147.96.1.162
Servidor SMTP	ucsmtp.ucm.es	147.96.1.163

La UCM ofrece como servicio básico a su comunidad:

- Webmail: Acceso al correo electrónico a través de la web utilizando el protocolo IMAP (http://webmail.ucm.es/).
- Filtro anti-spam: Se filtrarán los mensajes cuya dirección de origen, asunto, o contenido del mismo se encuentren en una lista "negra" proporcionada por la empresa Trend Micro y actualizada diariamente. Además, mediante un método heurístico, que va dando pesos a varias palabras para identificar tipos de contenidos y aquellos que superen el umbral definido son tratados como correo basura. Al no ser fiables al 100%, pudiendo dar falsos positivos, estos mensajes serán marcados con una etiqueta especial de forma que al principio del asunto del mensaje se añade "[POSIBLE SPAM]", permitiendo al usuario configurar su programa lector de correo para que borre, mueva o haga con dicho mensaje lo que quiera.
- Listas de Distribución: Servicio de listas de distribución LISTSERV. Este software permite crear, gestionar y sobre todo distribuir mensajes de listas de distribución de manera automatizada. Las listas de distribución se basan en el servicio de correo electrónico. Se crean listas sobre distintos temas de interés, para que los usuarios interesados se suscriban a ellas (enviando un correo electrónico o a través de la página Web). De esta manera, si se envía un correo electrónico a la lista de distribución, esta información llegará a todos los miembros de la lista suscritos. La lista es gestionada por uno o varios coordinadores cuya misión principal es hacer que se respetan las normas mínimas. No hay que confundir las listas con los grupos de noticias (newsgroups), que se utilizan para que un elevado número de usuarios debata sobre determinados temas, sin control sobre el contenido. Además utiliza otros protocolos y herramientas, diferentes al correo electrónico (con las limitaciones de filtrado, reglas para organizar los correos en carpetas o borrarlos, etc.).

Chapter 4. Diseño de la transición a IPv6 de la Red de Datos del Campus de Moncloa

4.1. Introducción

Una red corporativa puede elegir el despliegue de IPv6 de diferentes maneras, como está descrito en el documento "IETF Entreprise Scenarios". Nuestro objetivo es desplegar IPv6 en un entorno dual-stack donde los hosts que implementan servicios de aplicación (HTTP, SMTP, IMAP, etc) puedan optar por ofrecerlos sobre IPv6, al tiempo que dispositivos consumidores de servicios tengan acceso a servicios disponibles en la red IPv6 mundial.

Naturalmente, este proceso debe hacerse escaladamente, valorando en cada paso las implicaciones adyacentes como la compatibilidad, la estabilidad, la seguridad, etc. Un buen hito para este primer año sería soportar la conectividad nativa IPv6 permitiendo el uso del servicio HTTP sobre IPv6. Para ello es preciso apuntalar un mínimo necesario la actual infraestructura de red para que soporte IPv6. En años sucesivos se puede valorar el uso de protocolos y servicios exclusivos de IPv6 (Mobile IPv6, IPsec, Multicast, etc) muchos de los cuales están aun en fase de experimentación.

En la simulación que presentamos al efecto se puede observar como la inyección de IPv6 en el campus de Moncloa no ha de afectar a los actuales servicios de aplicación que actualmente se implementan en IPv4, para lo cual es necesario un profundo análisis de la infraestructura de Red de Datos de la Universidad Complutense. Esta tarea no hubiese sido posible sin la inestimable colaboración del personal del Centro de Proceso de Datos (CPD): Manuel Hernández Uerra, Luis Couto y Andrew.

4.2. Cambios necesarios para la migración

A continuación se enumeran algunos de los componentes de la actual red que habrían de verse modificados. Todos ellos han sido implementados con éxito en la simulación que hemos practicado.

- DNS: Los servidores internos de resolución de nombres (DNS) deben prepararse para poder resolver consultas AAAA, es decir las referidas a direcciones IPv6, sin dejar de resolver las correspondientes consultas a direcciones IPv4. A su vez, RedIris debería estar eventualmente preparado para derivar consultas DNS en IPv6 al dominio ucm.es. El primer punto está simulado en nuestro proyecto, incluso a nivel de DNS secundario, considerando ambos modos de transporte IPv4 e IPv6. Para el segundo, hay constancia de que RedIris-Madrid ya esta habilitado para este fin.
- Routing: El enrutado externo, tanto de IPv4 como de IPv6, esta estáticamente configurado en la conexión WAN hacia RedIris-Madrid. El enrutado interno IPv4 esta estáticamante configurado en el router CPD. Del mismo modo, está previsto el enroutado interno de IPv6, habiendo diseñado un plan de direccionamiento que se expondrá más adelante.

- Configuracion de Hosts: En la actualidad, los clientes IPv4 usan direcciones estáticas, salvo en la VLAN AIRE, que por sus propiedades características queda configurada por DCHPv6.
- Seguridad: La seguridad de una red forma parte de un estudio por separado que se debe considerar paralelamente a la implantación básica. En nuestro estudio, hemos asumido que, durante los primeros años, no se debe comprometer ningún servicio estratégico a IPv6. Esta política se puede complementar con modificaciones en el firewall de entrada haciendo que, por ejemplo, sólo determinados servicios como HTTP o FTP sean visibles al exterior, del mismo modo que se hace en la actualidad con IPv4 en el caso de la VLAN inalámbrica AIRE.

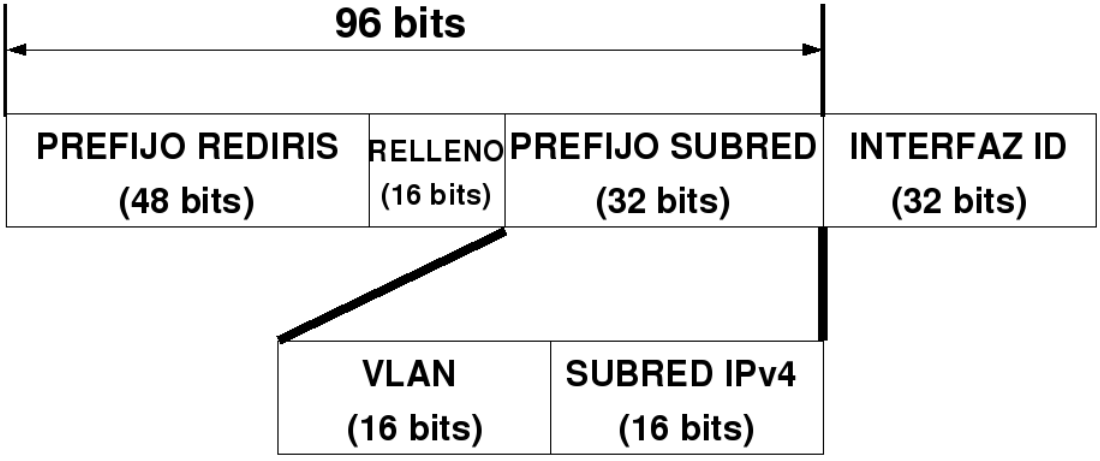
  En cuanto a tolerancia a fallos, cabe decir que este equipo, dentro de las posibilidades a las que puede acceder, no ha encontrado una implementación del protocolo VRRP para la versión IPv6, lo cual es un índice del papel secundario que ha de adoptar en los primeros años el nuevo protocolo.

- Gestión de red: Para la gestión de la red y la monitorización de los sistemas resulta imprescindible observar y gestionar el tráfico en IPv6. A fecha de hoy, hay que confirmar si los sistemas de monitorización del CPD están capacitados para ello. En todo caso, hay herramientas disponibles en el mundo del software libre, como MRTG, que implementan protocolos como SNMP capaces de gestionar el tráfico IPv6 y elaborar gráficas del tráfico entrante y saliente, que han sido utilizadas con éxito en otros proyectos como Euro6IX.

4.3. Modelo de direccionamiento IPv6

El modelo de direccionamiento que hemos diseñado es el siguiente:

Figure 4-1. Modelo de direccionamiento IPv6



La explicación de cada uno de los campos es la siguiente:

- Prefijo RedIris (48 bits): Aquí se incluye el identificador de subred IPv6 de 48 bits que le proporcione RedIris a la red Complutense. El hecho de que sean 48 bits se debe a que es el equivalente a redes de clase B en IPv6.
- Relleno (16 bits): Estos bits se dejan todos a 0 porque no tienen ninguna utilidad para nosotros.
- Prefijo de subred (32 bits): Este campo se divide a su vez en dos campos.
  - Identificador de VLAN (16 bits): Aquí se incluye el identificador correspondiente a la VLAN a la que pertenezca el host propietario de esta dirección. Esto facilita a los administradores saber de un vistazo a qué VLAN pertenece cada máquina.
  - Identificador de subred IPv4 (16 bits): Este identificador corresponde al tercer campo de la dirección IPv4 que tiene esta máquina. Hemos decidido incluirlo para que se pueda saber a simple vista a qué subred IPv4 corresponde una dirección Ipv6 y facilitar la gestión de la migración.
- Identificador de interfaz: Este campo es único para cada host dentro de una misma subred y corresponde al último campo de su dirección IPv4.

4.4. Estrategia de despliegue

El plan a seguir contempla los siguientes pasos:

- Obtener conectividad inicial. En nuestro caso, la conectividad es nativa IPv6 vía RedIris, desde su concesionaria regional RedIris-Madrid. Las negociaciones con el ente RedIris han de producirse por parte del personal del CPD, después de la aprobación técnica del proyecto y el visto bueno institucional del Vicerectorado de Innovación.
- Obtener un espacio de direccionamiento IPv6 y adoptar nuestro plan de direcciones.
- Desplegar una conectividad externa básica al tiempo que se asegura IPv6. Para cumplir este punto es necesaria la colaboración conjunta entre los equipos de RedIris-Madrid y el personal del CPD de la Universidad Complutense. Ya que la conectividad IPv6 no está disponible al mundo en general, no es previsible un entorno hostil vía IPv6. No obstante, una política conservadora de seguridad podría llevar a cerrar desde el firewall todos los servicios menos los más elementales.
- Despliegue de los servicios básicos de red DNS, routing y soporte a la configuración. El personal del equipo de Complu6IX ha adquirido experiencia con herramientas de software sobre IPv6 disponibles en Internet, con la configuración de equipos básica y con servicios de aplicación que pueden asesorar al personal del CPD cuando éste lo requiera.
- Despliegue de aplicaciones IPv6, gestión y soporte de servicios. Una red de datos tan grande como la de la UCM emplea sofisticados sistemas de balanceo de carga para los servicios que presta. Prevemos que, a corto y medio plazo, estos servicios no se verán afectados por la inyección de IPv6. A medio o largo plazo, debería considerarse la disponibilidad de estos sistemas en un entorno dual-stack.

Chapter 5. Prototipo

5.1. Simulador

5.1.1. Necesidad de la simulación

Una red de ordenadores está formado como mínimo por dos ordenadores y un medio de transmisión que los une, que puede ser un cable o un medio inalámbrico. Por lo tanto, si se quiere hacer pruebas en red disponer como mínimo de dos equipos y un cable. Pero con estos elementos tan solo se pueden hacer pruebas en una red muy básica. Posteriormente y conforme creciese el número de pruebas sería necesario adquirir más equipos y demás Hardware adicional (hubs, switches, routers...).

A partir de este punto existen dos opciones, comprar un Hardware mínimo que nos permita interactuar con una red real en la que realizar todas las pruebas necesarias o utilizar un simulador de redes. Para la realización del proyecto Complu6IX se ha utilizado la opción del simulador ya que de este modo no necesitamos una inversión de capital para empezar a trabajar sino que tan solo necesitamos un orndeador y un simulador de redes. Las simulaciones nos permiten construir escenarios e interactuar con ellos. Mediante la utilización de esta técnica podemos crear redes de gran complejidad de forma sencilla, barata y realizando los cambios en la configuración de la red que se deseen sin necesidad de instalar hardware adicional o modificar la configuración de routers o hosts.

Antes de explicar el escenario que construído para la red de datos de la UCM es necesario explicar algunos conceptos básicos relacionados con virtualización y simulación para entender los siguientes apartados.

5.1.2. User-Mode Linux

User-Mode Linux es un modo seguro de ejecutar procesos dentro del Sistema Operativo GNU/Linux llegando incluso a permitir la ejecución de distintas versiones de GNU/Linux dentro de una misma máquina. Está diseñado para permitir a desarrolladores experimentar distintas versiones de GNU/Linux sin poner en peligro la instalación del Sistema Operativo principal de nuestro computador. User-Mode Linux nos ofrece una máquina virtual que puede llegar a tener incluso más recursos hardware y software que el ordenador físico que la está ejecutando.

Para poder arrancar un Sistema Operativo del tipo GNU/Linux es necesario disponer de un sistema de ficheros que contiene los programas que ejecutará el ordenador y un kernel que se encarga de gobernar la ejecución de todos los procesos que está ejecutando el ordenador. La ventaja encontrada al utilizar el sistema de virtualización de User-Mode Linux es que todo el sistema de ficheros que tendrá nuestra máquina virtual está contenido en un único archivo en nuestro ordenador físico. Se puede asignar a la

máquina virtual el acceso al hardware que nosotros queramos que tenga, de modo que podemos realizar cualquier cambio en la configuración de la máquina virtual sin dañar nuestro ordenador físico, ni tan siquiera a su software.

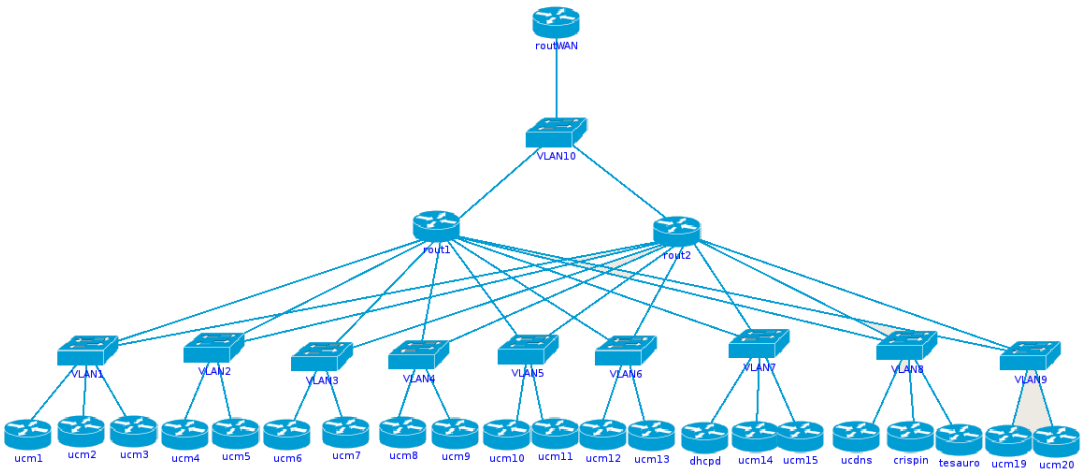
A partir de este momento ya podemos distinguir varios conceptos claves que se van a utilizar en el resto del documento.

- Ordenador anfitrión: Se trata del ordenador físico que contendrá la simulación. También nos referiremos a él como "host anfitrión".
- Máquina virtual: Se trata de un ordenador que no existe físicamente sino que ha sido creado para formar parte de un escenario de simulación. También nos referiremos a las máquinas virtuales como "hosts virtuales".

5.2. Descripción del escenario

Este capítulo está destinado a explicar la simulación que hemos diseñado y cómo se puede interactuar con ella para comprobar cómo quedaría la red de datos de la UCM tras implantar IPv6. El objetivo final de la simulación es comprobar cómo se puede implantar IPv6 en una red tan compleja como la establecida en la UCM sin perder ninguna de las funcionalidades existentes en IPv4. En la siguiente figura se puede ver el esquema de la simulación. Nuestro escenario está formado por 26 máquinas virtuales.

Figure 5-1. Esquema de la simulación



Tres de ellas son routers:

- rout1: Router principal de la UCM, se corresponde con el router LAN del CPD.
- rout2: Router secundario de la UCM, se corresponde con el router LAN del rectorado.
- routWAN: Tercer router de la UCM, se corresponde con el router WAN del rectorado, se encarga de dar salida a los datos a través de REDIRIS.

Las 23 máquinas virtuales restantes no tienen funciones de enrutado. Pero de ellas podemos destacar los servidores:

- ucdns: Servidor primario de DNS de la UCM, se corresponde con ucdns.sis.ucm.es.
- crispin: Servidor secundario de DNS de la UCM, se corresponde con crispin.sim.ucm.es.
- tesauro: Se corresponde con tesauro.sim.ucm.es. Hemos utilizado esta máquina virtual para albergar al resto de servidores de nuestra simulación: Web, Correo y FTP.
- dhcpd: Servidor de DHCP ubicado en la VLAN8 para distribuir direcciones IPv4 y/o IPv6 a los clientes que lo soliciten.

Los tres routers y los cuatro hosts utilizados como servidores son las máquinas virtuales más importantes de la simulación y sobre ellas haremos un especial seguimiento ya que mediante una configuración apropiada de las mismas conseguiremos implantar IPv6 sin perder todas las funcionalidades de IPv4.

Las máquinas virtuales que no son routers ni servidores son clientes que están distribuídos por todas las VLAN que hay en la UCM. Estos hosts están numerados desde ucm1 hasta ucm20 serán el objeto de nuestras pruebas. Desde ellos haremos pruebas de conectividad, leeremos la página Web de la UCM y más que se verán a continuación.

Table 5-1. Resumen de las IP's utilizadas en las pruebas de la simulación

Máquina vitual	IP	Valor
ucm1	IPv4	147.96.3.4/24
	IPv6	2001:720:1500:4200:100:3:0:4/96
ucm2	IPv4	147.96.3.14/24
	IPv6	2001:720:1500:4200:100:3:0:14/96
ucm3	IPv4	147.96.4.4/24
	IPv6	2001:720:1500:4200:100:4:0:4/96
dhcpd	IPv4	147.96.112.2/22
	IPv6	2001:720:1500:4200:506:112:0:2/96
ucm14	IPv4	ninguna

Máquina vitual	IP	Valor
	IPv6	ninguna
ucm15	IPv4	ninguna
	IPv6	ninguna
ucdns	IPv4	147.96.2.4/24
	IPv6	2001:720:1500:4200:600:2:0:4/96
crispin	IPv4	147.96.1.9/24
	IPv6	2001:720:1500:4200:600:1:0:9/96
tesauro	IPv4	147.96.1.38/24
	IPv6	2001:720:1500:4200:100:1:0:38/96
rout1	IPv4	147.96.3.1/24, 147.96.4.1/24, 147.96.112.1/22, 147.96.1.1/24, 147.96.2.1/24
	IPv6	2001:720:1500:4200:100:3:0:1/96, 2001:720:1500:4200:100:4:0:1/96, 2001:720:1500:4200:506:112:0:1/96, 2001:720:1500:4200:600:1:0:1/96, 2001:720:1500:4200:600:2:0:1/96
rout2	IPv4	147.96.3.2/24, 147.96.4.2/24, 147.96.112.2/22, 147.96.1.2/24, 147.96.2.2/24
	IPv6	2001:720:1500:4200:100:3:0:2/96, 2001:720:1500:4200:100:4:0:2/96, 2001:720:1500:4200:506:112:0:2/96, 2001:720:1500:4200:600:1:0:2/96, 2001:720:1500:4200:600:2:0:2/96

5.3. Pruebas sobre la simulación

Como se ha comentado con anterioridad, el principal objetivo de la simulación es mostrar que no se va a estropear nada de lo que funciona en estos momentos en la red de datos de la UCM por el hecho de introducir IPv6 en la misma. Del mismo modo, se va a demostrar que funciona la conectividad "dual stack". Durante la realización de las pruebas nos vamos a centrar en los cinco puntos que se exponen a continuación:

- VRRP.
- Conectividad IPv4 / IPv6.
- Enrutado IPv4 / IPv6.
- Servicios de aplicación duales: DNS y Web.

5.3.1. VRRP

Para comprobar el correcto funcionamiento de este protocolo de redundancia de routers es necesario ejecutar los siguientes comandos en cada uno de los routers de la simulación:

- Router primario: Este router (rout1) va a tener una prioridad más alta para convertirse en "MASTER" que el otro router (rout2). Por lo tanto, si rout1 está encendido va a convertirse en "MASTER" de forma automática.

```
rout1:~# ucarp -i eth1 -s 147.96.3.2 -v 10 -p complu6ix -P
             -a  147.96.3.1 -u vip-up.sh -d ./vip-down.sh
[INFO] Local advertised ethernet address is [fe:fd:00:00:10:01]
[WARNING] Switching to state: BACKUP
[WARNING] Switching to state: MASTER
[WARNING] Spawning [vip-up.sh eth1]
```

- Router secundario: Este router (rout2) va a servir como router de respaldo ("BACKUP") y tan solo se va a encargar del enrutado si el rout1 ha dejado de funcionar.

```
rout2:~# ucarp -b 5 -i eth1 -s 147.96.3.3 -v 10 -p complu6ix
              -a 147.96.3.1 -u vip-up.sh -d ./vip-down.sh
[INFO] Local advertised ethernet address is [fe:fd:00:00:11:01]
[WARNING] Switching to state: MASTER
[WARNING] Spawning [vip-up.sh eth1]
[WARNING] Switching to state: BACKUP
[WARNING] Spawning [./vip-down.sh eth1]
[WARNING] Preferred master advertised: going back to BACKUP state
```

Siendo vip-down.sh un script con el siguiente contenido:

```
rout2:~# cat vip-down.sh
#!/bin/sh
/sbin/ip -f inet addr del 147.96.3.1/24 dev eth1
```

y vip-up.sh otro script con el siguiente contenido:

```
rout1:~# cat vip-up.sh
#!/bin/sh
/sbin/ip -f inet addr add 147.96.3.1/24 dev eth1
```

El protocolo VRRP está diseñado de tal forma que si un router deja de funcionar otro router ocupará su lugar. Para comprobar dicho comportamiento podemos apagar el router primario para ver si el router secundario pasa a realizar correctamente las funciones de enrutado.

```
rout2:~# ucarp -b 5 -i eth1 -s 147.96.3.3 -v 10 -p complu6ix
             -a 147.96.3.1 -P -u vip-up.sh -d ./vip-down.sh
[INFO] Local advertised ethernet address is [fe:fd:00:00:11:01]
[WARNING] Switching to state: MASTER
[WARNING] Spawning [vip-up.sh eth1]
[WARNING] Switching to state: BACKUP
[WARNING] Spawning [./vip-down.sh eth1]
[WARNING] Preferred master advertised: going back to BACKUP state


[WARNING] Switching to state: MASTER
[WARNING] Spawning [vip-up.sh eth1]
```

Como se puede ver en el texto anterior, en cuanto se ha apagado el router primario y han pasado 5 segundos (tiempo establecido por el administrador) el router secundario ha pasado a ser "MASTER". Desde las máquinas virtuales de los clientes este cambio ha sido totalmente transparente y no han notado que se ha cambiado de router.

Lo último que queda por probar es la acción contraria, es decir, lo que sucedería si el router primario deja de funcionar y algún técnico soluciona el problema. En dicha situación, el router primario debería volver a ser el responsable de las funciones de enrutado.

```
rout1:~# ucarp -i eth1 -s 147.96.3.2 -v 10 -p complu6ix --neutral
            -a  147.96.3.1 -u vip-up.sh -d ./vip-down.sh
[INFO] Local advertised ethernet address is [fe:fd:00:00:10:01]
[WARNING] Switching to state: BACKUP
[WARNING] Switching to state: MASTER
[WARNING] Spawning [vip-up.sh eth1]
RTNETLINK answers: File exists
[WARNING] Putting MASTER DOWN (going to time out)
```

En el texto anterior se puede ver cómo en la última línea ("Putting MASTER DOWN") el nuevo "MASTER" se ha encargado de asumir las funciones de enrutado convirtiendo al anterior "MASTER" en "BACKUP". Este comportamiento se puede comprobar viendo la salida por pantalla de rout2:

```
rout2:~# ucarp -b 5 -i eth1 -s 147.96.3.3 -v 10 -p complu6ix
             -a 147.96.3.1 -P -u vip-up.sh -d ./vip-down.sh
[INFO] Local advertised ethernet address is [fe:fd:00:00:11:01]
[WARNING] Switching to state: MASTER
```

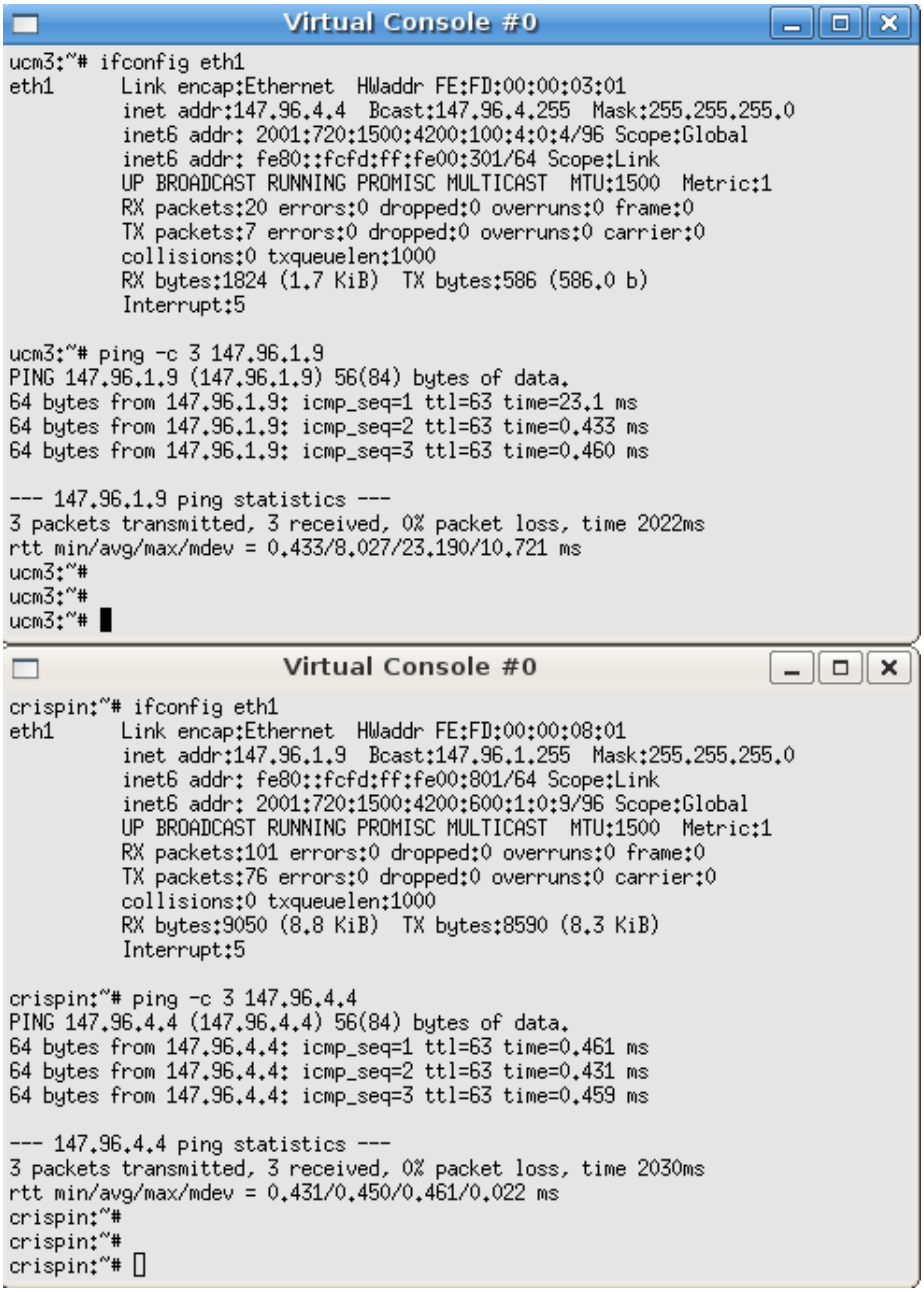
```
[WARNING] Spawning [vip-up.sh eth1]
[WARNING] Switching to state: BACKUP
[WARNING] Spawning [./vip-down.sh eth1]
[WARNING] Preferred master advertised: going back to BACKUP state


[WARNING] Switching to state: MASTER
[WARNING] Spawning [vip-up.sh eth1]


[WARNING] Switching to state: BACKUP
[WARNING] Spawning [./vip-down.sh eth1]
[WARNING] Preferred master advertised: going back to BACKUP state
```

La máquina rout2 ha detectado que el router primario ha vuelto a funcionar y automáticamente pasa a estado "BACKUP".

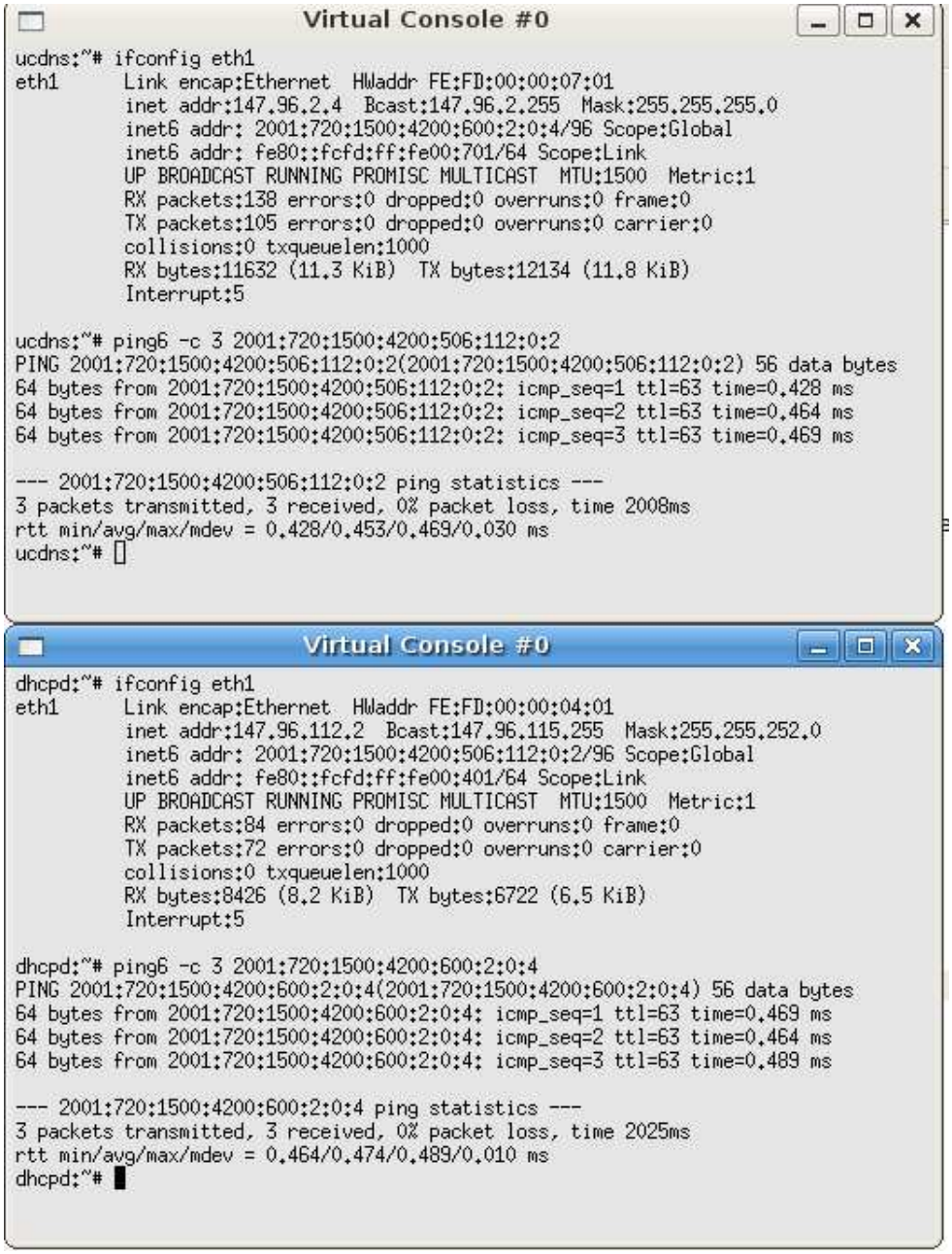
5.3.2. Conectividad IPv4 / IPv6

Para comprobar la correcta comunicación de las máquinas virtuales tanto en IPv4 como en IPv6 se van a mostrar una serie de pings entre varias máquinas:

1. Dos pings en IPv4 desde ucm3 a crispin y viceversa:



2. Dos pings en IPv6 desde ucdns a dhcpd y viceversa:



NOTA: Las pruebas anteriores se pueden repetir entre cualquiera de las máquinas virtuales de la simulación pero por simplicidad se ha resumido en dos ejemplos.

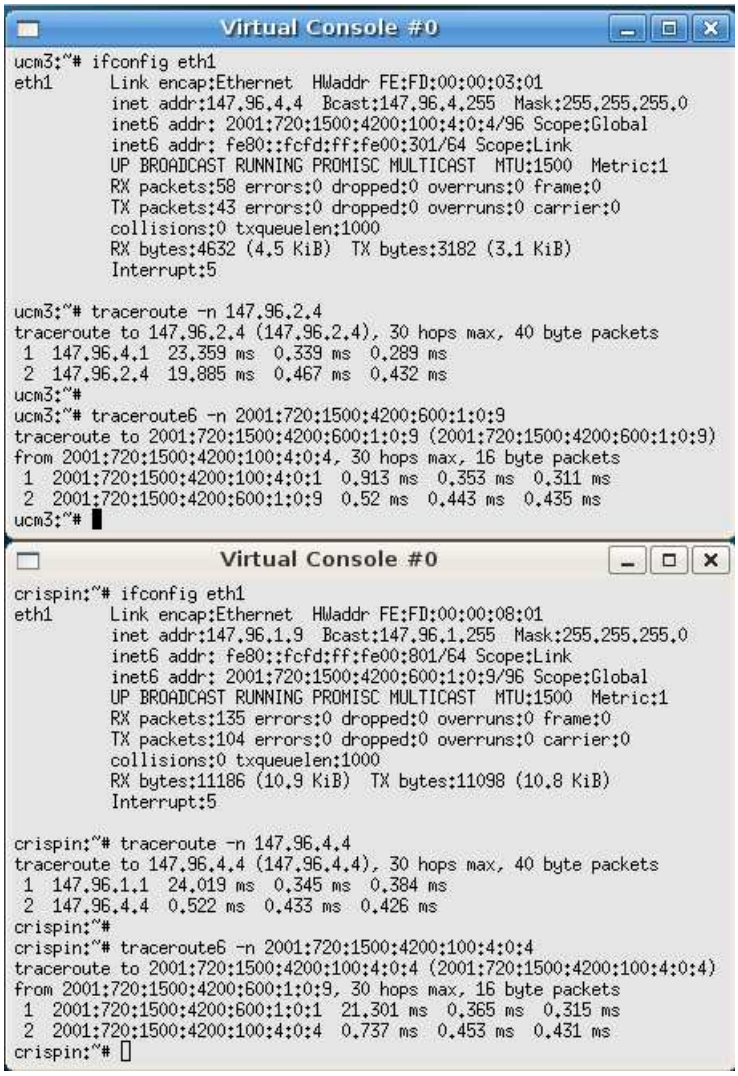
5.3.3. Enrutado IPv4 / IPv6

En esta simulación hay dos routers, rout1 es el router primario y rout2 es el router secundario. Sin embargo, como se ha explicado anteriormente, mediante el uso de VRRP cada uno de estos routers

puede ser el encargado del enrutado sin que el usuario note la diferencia.

Las pruebas que se van a mostrar a continuación tienen como objetivo comprobar que el enrutado de IPv6 se puede añadir al enrutado de IPv4 sin ningún problema para las comunicaciones IPv4. Además, mediante el enrutado diseñado para esta simulación las rutas IPv6 son análogas a las rutas IPv4. Es decir, si en una comunicación en IPv4 hay que pasar por el router para llegar al destino, en una comunicación IPv6 habrá que pasar igualmente por el router para llegar al destino.

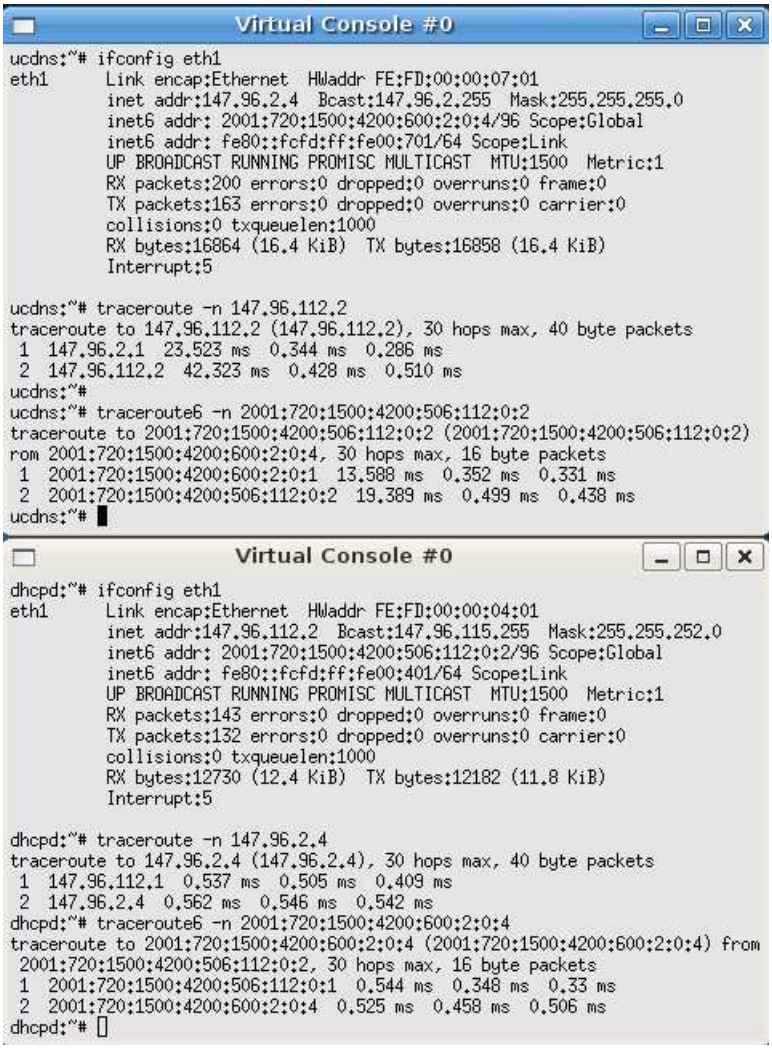
1. Dos traceroutes en IPv4 desde ucm3 a crispin y viceversa:



En esta captura se puede ver como para ir desde ucm3 a crispin utilizando IPv4 es necesario pasar por el router ya que se encuentran en subredes distintas (además de en VLAN distintas). De este modo, se puede ver que para realizar la misma conexión en IPv6 hay que pasar por el router de la

misma manera que en las conexiones IPv4.

2. Dos traceroutes en IPv6 desde ucdns a dhcpd y viceversa:



NOTA: Del mismo modo que con el ejemplo de la conectividad, las pruebas anteriores se pueden repetir entre cualquiera de las máquinas virtuales de la simulación pero por simplicidad se ha resumido en dos ejemplos.

5.3.4. Servicios de aplicación

En este capítulo se va a explicar cómo configurar los servicios básicos para conseguir que una red sea dual-stack IPv4 / IPv6.

5.3.4.1. DNS

Para conseguir conectividad con IPv6 el servicio más importante es el DNS, ya que permite tener una base de datos con las IP's y los nombres de las máquinas de una organización, tanto en IPv4 como en IPv4. El servidor DNS utilizado en la simulación es bind9.

La red de datos de la complutense es muy compleja como se ha explicado anteriormente, en una red tan grande es muy importante disponer de servicios redundantes. El servicio de DNS de la Universidad Complutense de Madrid dispone de un servidor primario "ucdns" y otro secundario "crispin". De tal modo que si ucdns deja de funcionar, crispin pasaría a ser el encargado de tratar todas las peticiones de nombres sin que el usuario final notase que está utilizando otro servidor de nombres (tal vez se notara la red un poco más lenta debido a que primero intenta conectar al servidor primario y si no lo consiguie entonces consulta al secundario).

Para conseguir que la simulación que presentamos sea lo más parecida a la real, se ha copiado este comportamiento. De tal modo que en la simulación hay dos servidores de nombres, el primario es "ucdns" y el secundario es "crispin". La correcta configuración de dichos servidores se consigue mediante los archivos de configuración `/etc/bind/named.conf` mostrados a continuación:

```
# FICHERO DE CONFIGURACION DE BIND (MASTER)

options {
    directory "/etc/bind/";
    listen-on-v6 {any;};
};

#
# DNS PARA IPV4
#

zone "ucm.es" in {
    type master;
    file "db.ucm.es";
};

zone "96.147.in-addr.arpa" in {
    type master;
    file "db.147.96";
};

#
# DNS PARA IPV6
#

zone "0.0.2.4.0.0.5.1.0.2.7.0.1.0.0.2.ip6.arpa" in {
    type master;
    file "db.147.96-v6";
};
```

```
zone "0.0.127.in-addr.arpa" in {
    type master;
    file "db.127.0.0";
};

zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa" in {
    type master;
    file "db.127.0.0-v6";
};

#
# DNS CACHE
#
zone "." in {
    type hint;
    file "db.cache";
};
```

En este archivo se pueden ver tres partes claramente diferenciadas. La primera tiene que ver con las opciones de resolución de nombres e IP en versión 4, la segunda está relacionada con la resolución de nombres e IP en este caso de vesión 6 y la tercera tiene que ver con opciones de dns cache. Mediante este archivo de configuración se indica al servidor donde se encuentran las bases de datos con la correspondencia nombre-IP de las máquinas de la red de datos de la UCM.

```
# FICHERO DE CONFIGURACION DE BIND DEL SERVIDOR SECUNDARIO (SLAVE)

options {
    directory "/etc/bind/";
    listen-on-v6 {any;};
};

#
# DNS PARA IPV4
#

zone "ucm.es" in {
    type slave;
    file "bak.ucm.es";
    masters {147.96.2.4;};
};

zone "96.147.in-addr.arpa" in {
    type slave;
    file "bak.147.96";
    masters {147.96.2.4;};
};

#
# DNS PARA IPV6
```

```
#

zone "0.0.2.4.0.0.5.1.0.2.7.0.1.0.0.2.ip6.arpa" in {
    type slave;
    file "bak.147.96-v6";
    masters {2001:720:1500:4200:600:2:0:4;};
};

zone "0.0.127.in-addr.arpa" in {
    type master;
    file "db.127.0.0";
};

zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa" in {
    type master;
    file "db.127.0.0-v6";
};

#
# DNS CACHE
#

zone "." in {
    type hint;
    file "db.cache";
};
```

En este archivo también encontramos las tres partes comentadas anteriormente, con la única salvedad que en lugar de indicar dónde se encuentra el archivo con la base de datos para transformar nombre en IP y viceversa, se encuentra la IP del servidor primario de nombres.

Llegados a este punto, tan solo resta mostrar un ejemplo de los tres ficheros con la base de datos para comprender de forma completa el funcionamiento de los servidores de nombres.

1. Fichero `db.ucm.es` cuyo objetivo es permitir al servidor resolver preguntas DNS del tipo nombre a IP:

   ```
   $TTL 24h
   ucm.es. IN SOA ucdns.sis.ucm.es. admin.ucm.es. (
       1  ; Serial
       3h ; Se refresca después de 3 horas
       1h ; Se reintenta después de una hora
       1w ; Expria tras una semana
       1h ) ; Cache negativo de 1 hora

   ;
   ; Servidores DNS
   ;
   ucm.es.     IN NS     ucdns.sis.ucm.es.
   ucm.es.     IN NS     crispin.sim.ucm.es.
   ```

```
;
; Direcciones para los nombres canónicos
;
localhost.ucm.es.      IN A       127.0.0.1
localhost.ucm.es.      IN AAAA    ::1
ucdns.sis.ucm.es.      IN A       147.96.2.4
ucdns.sis.ucm.es.      IN AAAA    2001:720:1500:4200:600:2:0:4
crispin.sim.ucm.es.    IN A       147.96.1.9
crispin.sim.ucm.es.    IN AAAA    2001:720:1500:4200:600:1:0:9
tesauro.sim.ucm.es.    IN A       147.96.1.38
tesauro.sim.ucm.es.    IN AAAA    2001:720:1500:4200:600:1:0:38
```

2. Fichero `db.147.96` cuyo objetivo es permitir al servidor resolver preguntas DNS del tipo IPv4 a nombre:

```
$TTL 24h
ucm.es. IN SOA ucdns.sis.ucm.es. admin.ucm.es. (
    1  ; Serial
    3h ; Se refresca después de 3 horas
    1h ; Se reintenta después de una hora
    1w ; Expria tras una semana
    1h ) ; Cache negativo de 1 hora

;
; Servidores DNS
;
96.147.in-addr.arpa.    IN NS    ucdns.sis.ucm.es.
96.147.in-addr.arpa.    IN NS    crispin.sim.ucm.es.

;
; Direcciones para los nombres canonicos
;
4.2.96.147.in-addr.arpa.    IN PTR    ucdns.sis.ucm.es.
9.1.96.147.in-addr.arpa.    IN PTR    crispin.sim.ucm.es.
38.1.96.147.in-addr.arpa.   IN PTR    tesauro.sim.ucm.es.
```

3. Fichero `db.147.96-v6` cuyo objetivo es permitir al servidor resolver preguntas DNS del tipo IPv6 a nombre:

```
$TTL 24h
ucm.es. IN SOA ucdns.sis.ucm.es. admin.ucm.es. (
    1  ; Serial
    3h ; Se refresca después de 3 horas
    1h ; Se reintenta después de una hora
    1w ; Expria tras una semana
    1h ) ; Cache negativo de 1 hora
```

```
;
; Servidores DNS
;
0.0.2.4.0.0.5.1.0.2.7.0.1.0.0.2.ip6.arpa.    IN NS    ucdns.sis.ucm.es.
0.0.2.4.0.0.5.1.0.2.7.0.1.0.0.2.ip6.arpa.    IN NS    crispin.sim.ucm.es.

;
; Direcciones para los nombres canonicos
;
4.0.0.0.0.0.0.0.2.0.0.0.0.0.6.0.0.0.2.4.0.0.5.1.0.2.7.0.1.0.0.2.ip6.arpa.
   IN PTR   ucdns.sis.ucm.es.
9.0.0.0.0.0.0.0.1.0.0.0.0.0.6.0.0.0.2.4.0.0.5.1.0.2.7.0.1.0.0.2.ip6.arpa.
   IN PTR   crispin.sim.ucm.es.
8.3.0.0.0.0.0.0.1.0.0.0.0.0.6.0.0.0.2.4.0.0.5.1.0.2.7.0.1.0.0.2.ip6.arpa.
   IN PTR   tesauro.sim.ucm.es.
```

Por último para que todo funcione es necesario indicar a cada máquina virtual cuál es la dirección IP de los servidores de nombres. Debido a que dichos servidores tienen IPv4 e IPv6 es necesario poner ambas en el fichero `/etc/resolv.conf` como se muestra a continuación:

```
ucm3:~# cat /etc/resolv.conf
# DNS de la ucm
nameserver 2001:720:1500:4200:600:2:0:4
nameserver 2001:720:1500:4200:600:1:0:9
nameserver 147.96.2.4
nameserver 147.96.1.9
```

En las pruebas que se van a mostrar a continuación van a estar involucradas las siquientes máquinas virtuales (se muestran sus IPs para comprobar el correcto funcionamiento de las pruebas):

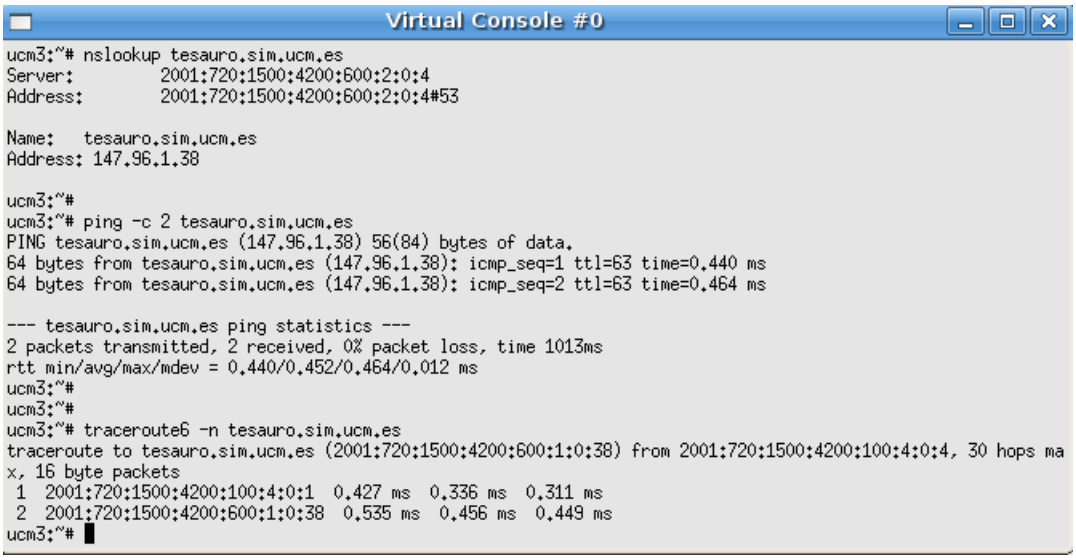
Table 5-2. Resumen de las IP's utilizadas en las pruebas de DNS

Máquina vitual	IP	Valor
ucm3	IPv4	147.96.4.4/24
	IPv6	2001:720:1500:4200:100:4:0:4/96
ucdns	IPv4	147.96.2.4/24
	IPv6	2001:720:1500:4200:600:2:0:4/96
crispin	IPv4	147.96.1.9/24
	IPv6	2001:720:1500:4200:600:1:0:9/96
tesauro	IPv4	147.96.1.38/24
	IPv6	2001:720:1500:4200:100:1:0:38/96

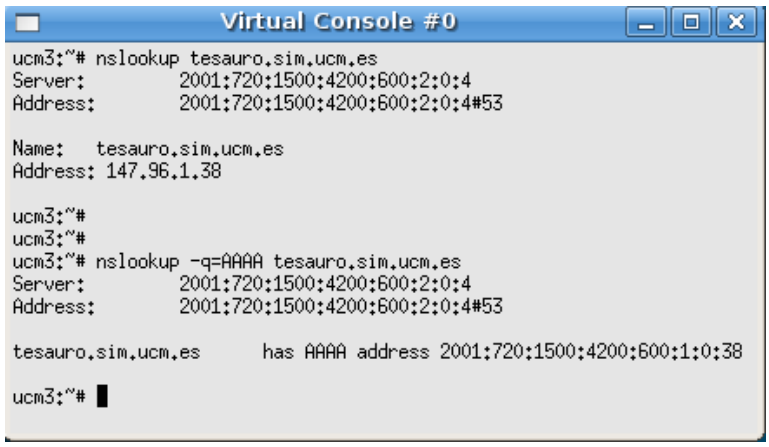
Para las pruebas mostradas a continuación se va a utilizar "ucm3" para realizar consultas DNS, se ha

elegido esta máquina virtual pero funciona con cualquiera de la simulación. Como se ha mostrado en los archivos de configuración anteriores se han incluído tres máquiquinas virtuales en la base de datos de nombres: crispin, ucdns y tesauro.

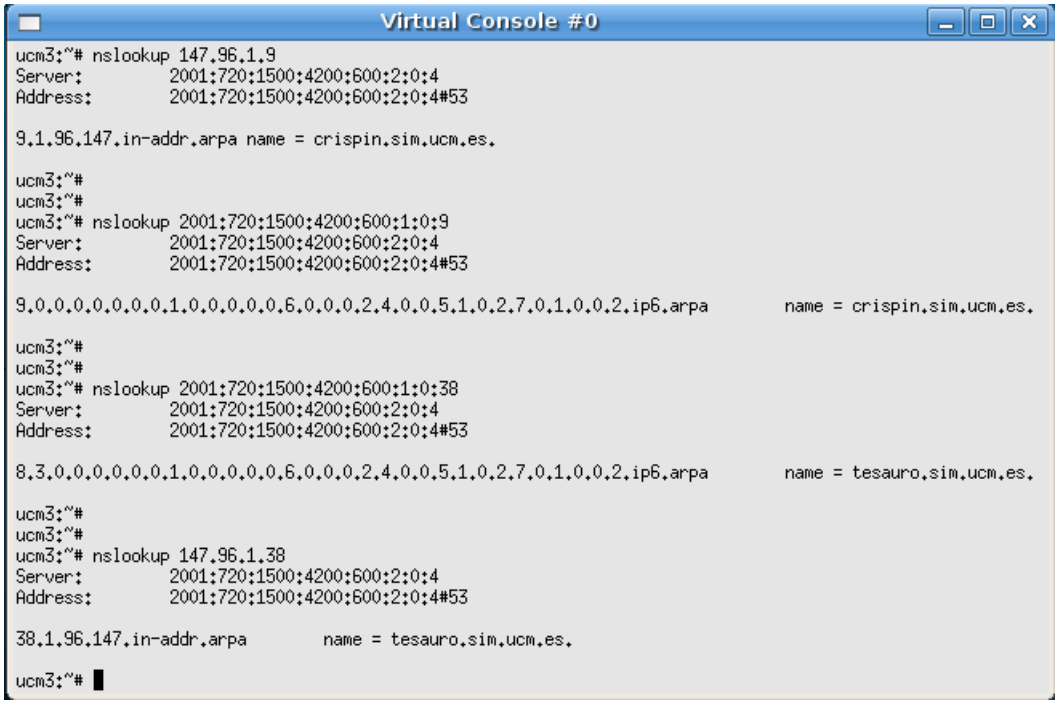
En la siguiente imagen se puede ver como utilizando *nslookup tesauro.sim.ucm.es* devuelve la IPv4 de la máquina consultada. Mediante la correcta configuarción de servicio de DNS se pueden hacer conexiones utilizando el nombres de la máquina en lugar de la IP ya que el servidor designado transformará el nombre a la IP correcta, es decir, si se hace una conexión IPv4 trasnforma tesauro.sim.ucm.es a 147.96.1.38 pero si se hace una conexión IPv6 trasnforma tesauro.sim.ucm.es a 2001:720:1500:4200:600:1:0:38. Este comportamiento se muestra a continuación:



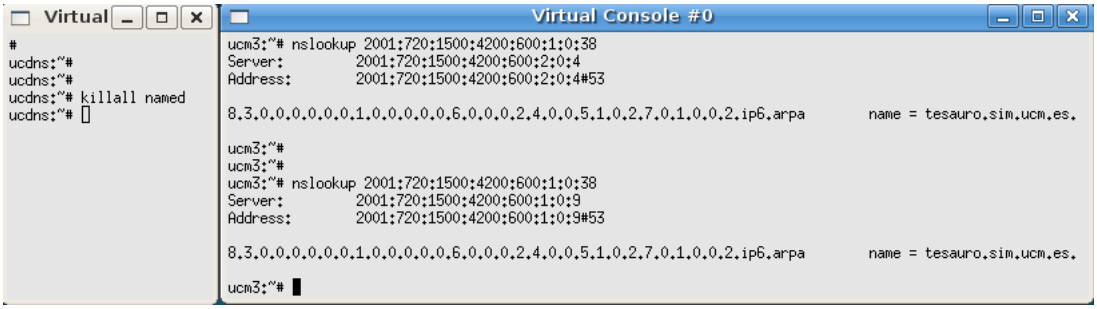
Si lo que se quiere consultar es la IPv6 de una máquina determinada se debe utilizar el comando *nslookup -q=AAAA tesauro.sim.ucm.es* como se muestra a continuación:



Es posible que lo que el usuario necesite sea conocer el nombre que está asociado a una IP tanto IPv4 como IPv6, para ello puede realizar las siguientes consultas:



Para terminar con las pruebas de DNS se va a mostrar qué es lo que sucede cuando el servidor primario de nombres deja de funcionar. Para ello se va a hacer una consulta DNS, posteriormente se va a parar el servidor bind9 de ucdns y se va a repetir la consulta DNS.



En esta imagen se puede ver como la primera consulta ha sido respondida por ucdns (la línea "Server" contiene la IPv6 2001:720:1500:4200:600:2:0:4). Después se ha parado el servidor bind9 de ucdns y se ha repetido la consulta pero la segunda consulta que ha sido respondida por cripin (la línea "Server" contiene la IPv6 2001:720:1500:4200:600:1:0:9).

5.3.4.2. Web

En la simulación se puede acceder a un servidor apache que está configurado para atender peticiones de conexión en IPv4 y en IPv6. Para configurar un apache de forma dual es necesario modificar el archivo `/etc/apache2/httpd.conf` de forma que quede como el que se muestra a continuación:

```
NameVirtualHost 147.96.1.38:80
NameVirtualHost [2001:720:1500:4200:600:1:0:38]:80

<VirtualHost 147.96.1.38:80>
  ServerName tesauro.sim.ucm.es
  DocumentRoot /var/www/apache2/v4htdocs
</VirtualHost>

<VirtualHost [2001:720:1500:4200:600:1:0:38]:80>
  ServerName tesauro.sim.ucm.es
  DocumentRoot /var/www/apache2/v6htdocs
</VirtualHost>
```

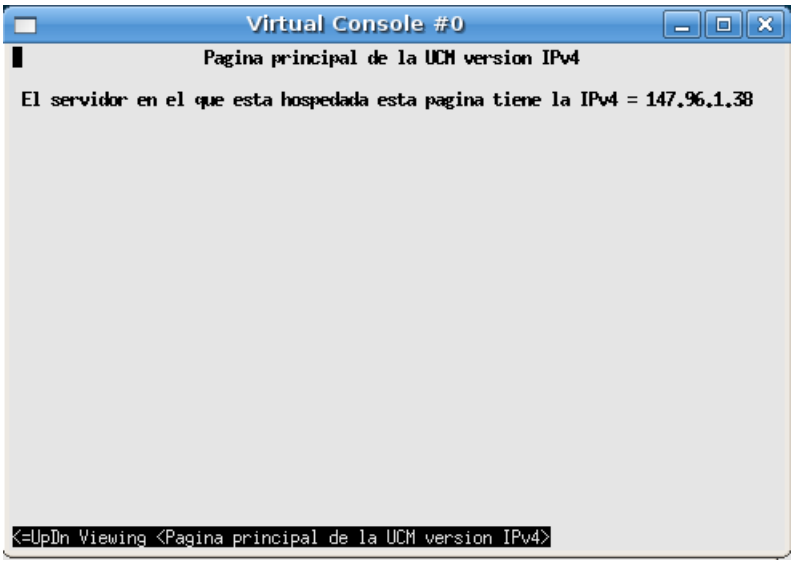
Este archivo muestra una configuración muy simple. En las dos primeras líneas indica cuáles son las IP's que se corresponden con los "VirtualHost" que se van a utilizar. Posteriormente, en cada etiqueta "VirtualHost" hay que completar el atributo "DocumentRoot" con el directorio en el que se encuentran los archivos que deseamos mostrar al usuario que se conecta por IPv4 o por IPv6.

Una vez que el servidor apache ha sido lanzado de forma satisfactoria tan solo hay que seguir los siguiente pasos para leer una página Web en IPv4 o en IPv6:

1. IPv4: Hay que escribir en una consola lo siguiente:

   ```
   w3m 147.96.1.38
   ```

   El resultado se puede ver en la captura mostrada a continuación:



2. IPv6: Hay que escribir en una consola lo siguiente:

   ```
   w3m [2001:720:1500:4200:600:1:0:38]
   ```

   El resultado se puede ver en la captura mostrada a continuación:

5.3.4.3. DHCP

Aparte de direccionamiento estático, una parte de la red de datos de la Complutense utiliza DHCP para otrogar direcciones IP. Esta técnica también se puede observar en la simulación. Para ello se ha incluído en la VLAN7 un servidor DHCP de IPv4 y de IPv6 junto con dos máquinas virtuales a las que no se ha dotado de dirección IP. Con la prueba que se va a mostrar a continuación se va a comprobar que en una red se pueden dar direcciones IPv4, IPv6 o ambas de forma dinámica. Del mismo modo, se verá que en una misma máquina se puede tener instalado un servidor DHCPv4 y otro DHCPv6 sin que haya ningún problema.

Para conseguir otorgar direcciones IP es necesario configurar un servidor DHCP para la versión IPv4 y otro para la versión IPv6. Hay que seguir los siguientes pasos:

1. IPv4: Se ha utilizado la versión de dhcpd ISC para Debian. Es necesario editar el archivo `/etc/dhcp3/dhcpd.conf` como se muestra a continuación:

```
#
# configuration file for ISC dhcpd for Debian
#

ddns-update-style none;

option domain-name "ucm.es";

option domain-name-servers 147.96.2.4, 147.96.1.9;

default-lease-time 600;

max-lease-time 7200;

log-facility local7;

subnet 147.96.112.0 netmask 255.255.252.0 {

   range 147.96.112.4 147.96.113.250;
   option broadcast-address        147.96.115.255;
   option domain-name              "ucm.es";
   option domain-name-servers      147.96.2.4;
   option routers                  147.96.112.1;

}
```

   Mediante este simple archivo de configuración se puede establecer el tiempo máximo que puede permanecer una IP, el rango IP's que se van a entregar a los clientes, los servidores de DNS que utilizarán los clientes así como el router y la dirección broadcast.

2. IPv6: Se ha utilizado la versión dibbler-dhcpd. Es necesario editar el archivo `/etc/dibbler/server.conf` como se muestra a continuación:

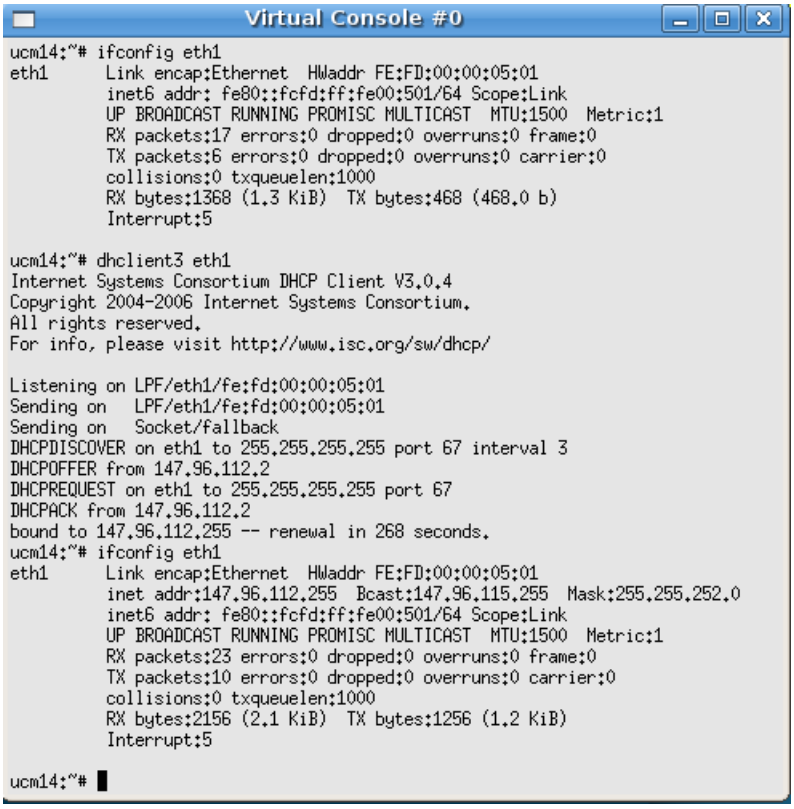
```
#
# configuration file for server.conf (dibbler-dhcpd)
#
experimental

iface eth1
{
T1 1800
T2 2000
prefered-lifetime 3600
valid-lifetime 7200


class {
  addr-params 96
  pool 2001:720:1500:4200:506:112::1 -
              2001:720:1500:4200:506:112:ffff:ffff
}

option dns-server 2001:720:1500:4200:600:2:0:4 ,
              2001:720:1500:4200:600:1:0:9
}
```

Este archivo de configuración es análogo al mostrado anteriormente con la salvedad de que todas las opciones son para IPv6, se puede ver cómo se establece el tiempo máximo de permanencia de una IPv6 en los clientes y cómo se establece el "pool" de direcciones. Dicho "pool" es el rango de direcciones que se van a dar a los clientes que lo soliciten. Igual que en el servidor de IPv4 se puede acompañar a la dirección del host con las direcciones de los servidores DNS.

Dado que en la simulación utilizamos direccionamiento IPv6 con una máscara de red de 96 bits, es necesario que el servidor reparta las direcciones con los 96 bits de prefijo de red (por defecto son 64). Para ello hay que utilizar la opción de "prefix-delegation", dicha técnica permite al cliente configurar de forma adecuada su nueva dirección IPv6. Para poder utilizar esta técnica es necesario incluir la palabra "experimental" al principio del fichero y la línea "addr-params 96" dentro de la sección "class".

NOTA: Mientras se estaba escribiendo este documento la opción "prefix-delegation" estaba en fase de pruebas y ese es el motivo de incluir la palabra experimental en el archivo de configuración.

Para la realización de esta prueba, en primer lugar se va a ejecutar el cliente de DHCPv4 en la máquina virtual ucm14. De este modo ya se prodrían realizar conexiones con IPv4.

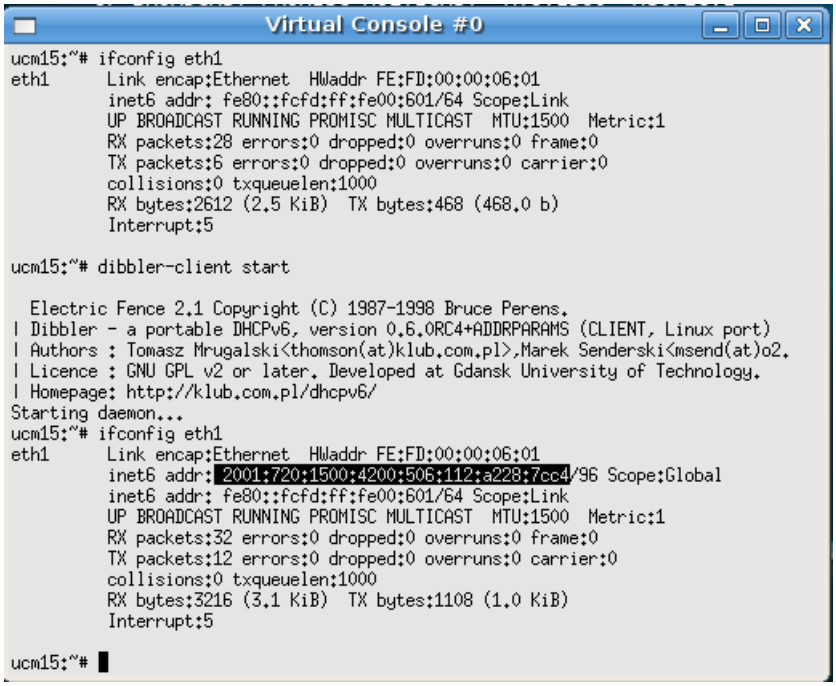


En la imagen anterior se puede ver como inicialmente la máquina ucm14 no tiene ni dirección IPv4 ni dirección IPv6 pero después de ejecutar *dhclient3 eth1* ya dispone de dirección IPv4.

Si, por el contrario, se desea dotar de direcció IPv6 a una máquina virtual se puede ejecutar el cliente DHCPv6 mediante el comando *dibbler-client start*. El archivo de configuración necesario para que funcione correctamente debe encontrarse en `/etc/dibbler/client.conf` con el siguiente contenido:

```
log-mode short
experimental
log-level 8
iface eth1 {
  T1 1800
  T2 2000
  prefered-lifetime 3600
  valid-lifetime 7200
  ia {
    addr-params
  }
  option dns-server
}
```

Al ejecutar el cliente dibbler se obtiene el siguiente resultado:



En esta imagen se puede ver como inicialmente la máquina ucm15 no dispone de dirección IPv4 ni IPv6 y tras ejecutar el comando *dibbler-client start* ya tiene una dirección IPv6 *2001:720:1500:4200:506:112:a228:7cc4* asignada.

Del mismo modo, es importante comprobar el archivo `/etc/resolv.conf`. En el archivo de configuración del cliente y servidor dibbler se ha establecido que se debe anunciar también a los clientes los servidores de nombres DNS. Dicho archivo en la máquina virtual ucm15 y uml14 inicialmente está vacío y después de ejecutar el cliente dibbler pasa a tener el siguiente contenido:

```
ucm15:~# cat /etc/resolv.conf
nameserver 2001:720:1500:4200:600:2:0:4
nameserver 2001:720:1500:4200:600:1:0:9
```

Llegados a este punto tan solo resta mostrar que es posible ejecutar ambos clientes en versión IPv4 e IPv6 en una máquina. Para ello se han ejecutado los dos clientes DHCP en las máquina virtuale uml15. El resultado se muestra a continuación:

En la captura de pantalla anterior se puede ver como es posible ejecutar un cliente y después otro obteniendo como resultado que la máquina virtual pasa a tener dos nuevas direcciones IP una IPv6 y la otra IPv4.

5.4. Limitaciones del simulador

El simulador utilizado tiene algunas limitaciones que se muestran a continuación.

1. Limitación de subredes a 64 y número de nodos de la simulación. Teóricamente no hay límite de nodos en la simulación ya que se puede crear un escenario en el archivo xml con todas las máquinas virtuales que el usuario desee, sin embargo, cada máquina virtual tarda aproximadamente un minuto en arrancar completamente. De este modo, durante la duranción del proyecto Complu6IX no se han ejecutado simulaciones con más de 15 máquinas virtuales.
2. Limitación en cuanto al Sistema Operativo de las máquinas virtuales de la simulación. Todas las máquinas de la simulación deben ser GNU/Linux, en cualquiera de sus versiones. En la realización del proyecto Complu6IX se ha decidido que todas las máquinas virtuales tengan como Sistema

Operativo Debian (testing), se podrían haber incluído más versiones de GNU/Linux en la simulación pero no se habría notado en el resultado ya que el funcionamiento de cualquiera de las versiones de GNU/Linux se comporta de forma similiar ante las conexiones IPv4 e IPv6. El verdadero problema de esta limitación es que no se han podido incluir en la simulación máquinas con otros Sistemas Operativos propietarios.

3. Limitación sobre el tipo de máquinas virtuales de la simulacion. Esta limitación está intimamente unida a la anterior, debido a que tan solo se pueden simulas máquinas con GNU/Linux no se pueden incluir impresoras, videocámaras u otros dispositivos que se puedan conectar en red.
4. Limitación en cuanto a la simulación de la capa física de red. No se puede simular fibra óptica, repetidores, puentes transparentes, etc. Tan solo se puede mostrar en la simulación redes LAN tipo Ethernet o conexiones punto a punto.

Chapter 6. Conclusiones

Durante la realización de este proyecto los miembros de Complu6IX hemos experimentado con las distintas posibilidades que ofrece IPv6. Inicialmente, el proyecto avanzaba lentamente debido a toda la información que era necesario asimilar antes de hacer las simulaciones y las pruebas sobre las simulaciones. Sin embargo, a medida que la toma de contacto con IPv6 se hacía más estable (experimentos con tunnel-brokers, configuración de aplicaciones para que soporten IPv6, etc.) hemos descubierto que la tecnología IPv6 está madura y preparada para ser implantada en cualquier organización.

A lo largo de estos últimos ocho meses hemos trabajado de forma conjunta con Alicia Melado, Silvia Corredor y Sofía Muñoz que se han encargado de mejorar la herramienta de simulación que se ha utilizado para realizar todas las pruebas. En colaboración con Alicia, Silvia y Sofía se ha creado la siguiente página Web donde se ha ido publicando información sobre el desarrollo del proyecto: *http://complu6ix.ucm.es/*.

Por último es necesario recordar que la simulación realizada por lo miembros del proyecto Complu6IX ha intentado ser lo más fiel a la realidad. Para ello ha sido importantísima la colaboración con el Centro de Proceso de Datos de la UCM que nos ha facilitado información sobre la infraestructura de la red de datos de la Complutense además de haberse reunido con nosotros en varias ocasiones para comprobar el avance del proyecto.

UNIVERSIDAD COMPLUTENSE DE MADRID.

CENTRO DE PROCESO DE DATOS.

Don Manuel Hernández Urrea, Director del Área de Seguridad, Redes e Infraestructura de los Servicios Informáticos, tras evaluar los resultados del proyecto denominado

COMPLU6IX: TRANSICIÓN A IPv6 DE LA RED DE DATOS DE LA UNIVERSIDAD COMPLUTENSE

integrado por los alumnos:

Alberto Abián Belmonte, Jaime Frutos Morales, Javier Santa María Reinoso, Alicia Melado Corral, Sofía Muñoz Vélez y Silvia Corredor Sanz

y dirigidos por el profesor **Rafael Martínez Torres**,

manifiesta el interés, utilidad y aprovechamiento del mismo para la Comunidad Universitaria.

Dicho proyecto examina la compatibilidad y coexistencia de las tecnologías IPv4 e IPv6 en el seno de la Red de Datos de la Complutense, presentando al efecto una simulación del escenario en el que se recrean los principales elementos de la infraestructura de la Red (*routers*, dispositivos de *switching*, principales servicios de aplicación, etc.) y un plan de direccionamiento a seguir ajustado a las características propias de la misma en coordinación con el ente RedIRIS, proveedor de conectividad a la Universidad.

En vista de lo satisfactorio de los resultados, en los próximos meses se prevé la apertura de un nuevo grupo de trabajo que lleve a la implementación escalada de IPv6 en la UCM.

Manuel Hernández Urrea.
Madrid, 18 de Mayo 2007.

Appendix A. Apéndice

Este apéndice trata :

- Manual de instalación de VNUML
- Manual de uso de VNUML

A.1. Manual de instalación de VNUML

A.1.1. Instalación del simulador

Hay distintas formas de instalar VNUML como se explica en la página web:

`http://www.dit.upm.es/vnumlwiki/index.php/Installation`

Dependiendo de la versión de GNU/Linux que esté instalada en el sistema es más recomendable utilizar uno u otro. Para la realización del proyecto Complu6IX se ha utilizado una instalación desde las fuentes. El hecho de utilizar instalación desde fuentes nos ha sido muy últil ya que así todos los miembros del grupo tienen la misma versión del software independientemente de la versión de GNU/Linux que utilizara cada uno.

La versión utilizada durante toda la realización del proyecto ha sido la 1.7.0-1 con fecha del 27/07/2006. Aproximadamente con el 60% del proyecto terminado se lanzó la versión 1.8 del simulador. A pesar de las mejoras de la nueva versión, se decidió continuar con la versión 1.7.0-1 debido a que había algunos cambios en la DTD que harían necesario modificar los escenarios de la simulación.

Lo primero es descargar el archivo que contiene el código fuente de la aplicación:

`http://sourceforge.net/project/showfiles.php?group_id=113582&package_id=123470`

Hay que descargar el archivo: vnuml-1.7.0-1.tar.gz. Una vez descargado tan solo hay que escribir los siguientes comandos:

```
bash$ tar xvfz vnuml-1.7.0-1.tar.gz
bash$ cd vnuml-1.7.0-1
bash$ sudo ./configure --with-expat=yes --with-build_modules=yes
             --with-tun=/dev/net/tun
bash$ make</screen>
```

```
bash$ sudo make install
```

Antes de ejecutar el configure, es necesario instalar algunos paquetes extra para que la aplicación funcione de forma correcta. Dichos paquetes son los que listan a continuación:

- tunctl, uml_mconsole y uml_switch: En distribuciones de GNU/Linux basadas en Debian se encuentran en el paquete uml_utilities, para otro tipo de distribución los fuentes se encuentran en http://user-mode-linux.sourceforge.net/dl-sf.html
- brctl: Si se van a utilizar redes virtuales con la opción de virtual-bridge es necesario instar el paquete bridge-utils.
- vconfig, vlan: Si se van a utilizar VLAN en las simulaciones.
- screen: Utilizado para poder interactual con las máquinas virtuales.

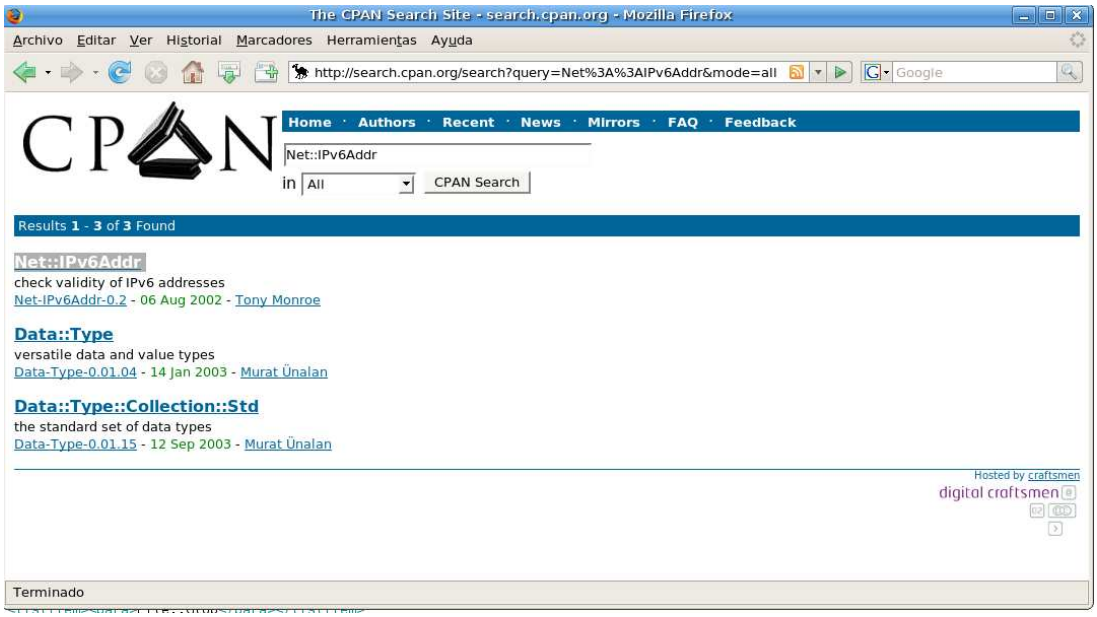
Del mismo modo, es necesario descargar e instalar los siguientes módulos Perl:

- Error
- Exception::Class
- XML::DOM
- XML::Checker
- XML::RegExp
- XML::Parser
- XML::Parser::PerlSAX
- NetAddr::IP
- Net::Pcap
- IO::Socket
- Term::ReadKey
- Net::IPv6Addr
- Math::Base85
- Net::IPv4Addr
- File::Glob

Hay dos formas de hacerlo:

- Ir la página web http://search.cpan.org/, buscar el módulo en concreto y descargarlo.

Figure A-1. Búsqueda en la web de CPAN



Una vez descargado tan solo hay que escribir los siguientes comandos:

```
bash$ tar xvfz Net-IPv6Addr-0.2.tar.gz
bash$ cd Net-IPv6Addr-0.2
bash$ perl Makefile.PL
bash$ make
bash$ sudo make install
```

Es conveniente, pero no obligatorio, antes de hacer "make" hacer make test. El resultado de esta orden es imprimir por pantalla el resultado de múltiples pruebas realizadas sobre el módulo que se desea instalar. Si todas las pruebas devuelven "ok" es que todo ha ido bien, si alguna no devuelve "ok" es posible que sea necesario instalar alguna dependencia extra. En la instalación del módulo Net::IPv6Add el resultado de "make test" es el siguiente:

```
bash$ make test
PERL_DL_NONLAZY=1 /usr/bin/perl "-MExtUtils::Command::MM" "-e"
 "test_harness(0, 'blib/lib', 'blib/arch')" t/*.t
t/base85........ok
t/chkip.........ok
t/compressed....ok
t/ipv4..........ok
t/ipv4comp......ok
t/new...........ok
t/parse.........ok
t/preferred.....ok
t/string........ok
All tests successful.
Files=9, Tests=287,  2 wallclock secs ( 0.86 cusr + 0.11 csys =  0.97 CPU)
```

- Utilizar la shell cpan: Hay un programa que contiene una pequeña shell que se encarga de descargar e instalar los módulos Perl que sean necesarios. Para utilizarlo hay que teclear el siguiente comando (es necesario ejecutarlo como superusuario o root):

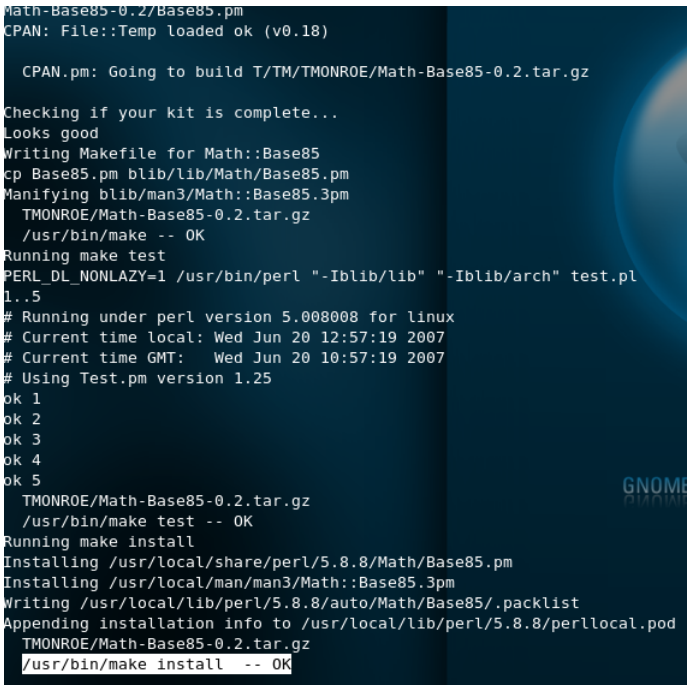
  ```
  bash# cpan
  ```

  ```
  cpan[1]> install Math::Base85
  ```

  Este método es más cómodo que el anterior debido ya que tan solo se necesita escribir la palabra "install" seguida del nombre del módulo y el programa se encarga de conectarse a la base de catos CPAN, descargar el archivo, compilarlo, pasar los tests e instalarlo. Es importante reseñar que es imprescindible ejecutar el programa cpan con permisos de root ya que si no es así no permiritá instalar el módulo elegido. Del mismo modo, si el módulo elegido tiene alguna dependencia, el programa instalará dicha dependencia de forma transparente para el usuario. Sin embargo, en ocasiones da algún error o no encuentra el módulo deseao y, en ese caso, es necesario hacer una instalación manual (por eso se ha explicado anteriormente).

  Para comprobar que un módulo se ha instalado correctamente, hay que fijarse en las últimas líneas que el programa cpan imprime por pantalla. La última palabra que aparezca debe ser OK como se muestra a continuación:

Figure A-2. Instalación en la shell CPAN



A.1.2. Instalación del kernel y el filesystem

Para que las simulaciones funcionen es necesario tener un sistema de ficheros en el disco duro del equipo anfitrión que tenga instalado el sistema operativo y los programas que se ejecutarán al lanzar la simulación. El sistema operativo que vamos a simular es Debian (testing). Dado que UML está diseñado para simular distintas versiones de GNU/Linux tanbién será necesario in kernel que permitirá arrancar el sistema operativo virtual. Estos componentes se pueden descargar de la siguiente página Web: http://www.dit.upm.es/vnumlwiki/index.php/Download.

- El kernel elegido durante la realización del proyecto Complu6IX ha ido cambiando conforme aparecían las distintas versiones en dicha página Web pero al final se optó por el kernel linux-2.6.18.1-bb2-xt-1m
- Como sistema de ficheros inicialmente se utilizó la versión 0.4.0 (ext2), pero fue sustituída inmediatamente en cuanto en la página web se actualizó al sistema de ficheros 0.4.1 (ext3) debido a las numerosas ventajas que conlleva utilizar un sistema de ficheros con ext3 frente a uno ext2.

Para el correcto funcionamiento del kernel es necesario descargarlo y descomprimirlo en la carpeta `/usr/local/share/vnuml/kernels/`, del mismo modo, el sistema de ficheros debe ser descargado y descomprimido en la carpeta `/usr/local/share/vnuml/filesystems/`.

Una vez completado el paso anterior, es necesario instalar los módulos del kernel descargado dentro del sistema de ficheros que se va a utilizar en las simulaciones, para ello hay que seguir los siguientes pasos:

1. Montar el sistema de ficheros:

   ```
   mount sistema_de_ficheros /mnt/uml -o loop
   ```

2. Descomprimir el archivo con los módulos en la carpeta `/lib/modules/` dentro del sistema de ficheros

   ```
   cd /mnt/uml/lib/modules ; tar xfv /root/modules-2.6.18.1-bb2-xt-1m.tar
   ```

   (Suponiendo que el archivo `modules-2.6.18.1-bb2-xt-1m.tar` se encuentra en el directorio `/root/`)

A.1.3. Configuración del sistema de ficheros

En este apartado se va a explicar cómo se redimensiona, como se actualiza y cómo se instalan nuevos programas el sistema de ficheros para que se pueda utilizar conforma a las necesidades del usuario final.

Es necesario señalar que el sistema de ficheros descargado de la página de VNUML es un archivo que contiene el sistema de ficheros completo con una Debian testing cuya capacidad es de 750MB aproximadamente.

- Es posible que el usuario final necesite instalar algún programa y que no tenga suficiente espacio, para redimensionar el sistema de ficeros es necesario seguir los siguientes pasos:

  1. Verificar el estado del sistema de ficheros:

     ```
     bash# e2fsck -f filesystem
     e2fsck 1.39 (29-May-2006)
     Paso 1: revisando nodos i, bloques y tamaños
     Paso 2: revisando la estructura de directorios
     Paso 3: revisando la conectividad del directorio.
     Paso 4: revisando las cuentas de referencia
     Paso 5: revisando el resumen de información del grupo
     filesystem: 43022/189504 files (8.5% non-contiguous), 667805/768000 blocks
     ```

  2. Establecer el nuevo tamaño deseado: (parámetro seek, en este caso se le está dando un tamaño de 1.4GB)

     ```
     bash# dd if=/dev/zero of=filesystem bs=1 count=1 seek=1400M conv=notrunc
     1+0 records in
     1+0 records out
     un byte (1 B) copiado, 0,000138 segundos, 7,2 kB/s
     ```

  3. Modificación del archivo para que tenga el nuevo tamaño:

     ```
     bash# resize2fs -p filesystem
     resize2fs 1.39 (29-May-2006)
     Resizing the filesystem on filesystem to 1433600 (1k) blocks.
     Se comienza el paso 1 (máx = 81)
     Extendiendo la tabla de nodos iXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
     Se comienza el paso 2 (máx = 2271)
     Reubicando bloques            XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
     Se comienza el paso 3 (máx = 94)
     Revisando la tabla de nodos i XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
     Se comienza el paso 5 (máx = 10)
     Moviendo la tabla de nodos i  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
     The filesystem on filesystem is now 1433600 blocks long.
     ```

  4. Revisión para comprbar el correcto estado del archivo:

     ```
     bash# e2fsck -f filesystem
     e2fsck 1.39 (29-May-2006)
     Backing up fichero de transacciones nodo i bloque information.
     ```

```
Paso 1: revisando nodos i, bloques y tamaños
Paso 2: revisando la estructura de directorios
Paso 3: revisando la conectividad del directorio.
Paso 4: revisando las cuentas de referencia
Paso 5: revisando el resumen de información del grupo

filesystem: ***** EL SISTEMA DE FICHEROS FUE MODIFICADO *****
filesystem: 43022/352800 files (8.5% non-contiguous), 688416/1433600 blocks
```

- Instalación de software extra en el sistema de ficheros: Todos los programas que se quieran ejecutar en los hosts virtuales deben encontrarse instalados en el sistema de ficheros. Para instalarlos hay que seguir alguno de estos dos pasos:
  - Instalación desde código fuente:

    ```
    bash# mount -o loop filesystem /mnt
    bash# cp archivo_codigo_fuente.tar.gz /mnt/root/
    bash# chroot /mnt
    bash# cd /root
    bash# tar xvfz archivo_codigo_fuente.tar.gz ; cd codigo_fuente
    bash# ./configure
    bash# make
    bash# exit
    bash# umount /mnt
    ```

    Con los comandos puestos anteriormente se monta el sistema de ficheros en un directorio de nuestro ordenador, a continuación, se copia el archivo que contiene el código fuente al sistema de ficheros para posteriormente descomprimirlo e instalarlo. Como se puede ver se hace un "chroot" el directorio en el que se ha montado el sistema de ficheros, lo cual permite reubicar el directorio root a un nuevo directorio (en nuestro caso `/mnt/`).

  - Instalación de nuevo software utilizando apt-get:

    ```
    bash# mount -o loop filesystem /mnt
    bash# chroot /mnt
    bash# gpg --keyserver wwwkeys.eu.pgp.net --recv-keys A70DAF536070D3A1
    bash# apt-key add /root/.gnupg/pubring.gpg
    bash# apt-get update
    bash# apt-get install nuevo_programa
    bash# exit
    ```

- Actualización del sistema operativo contenido en el sistema de ficheros: Dado que el sistema operativo contenido en el sistema de ficheros es Debian, la actualización es muy simple, tan solo hay que seguir los siguientes pasos:

```
bash# mount -o loop filesystem /mnt
bash# chroot /mnt
bash# apt-get update
bash# apt-get upgrade
bash# exit
```

NOTA: Es posible que al hacer un apt-get no se conecte a internet, para solucionar este problema hay que revisar el fichero `/etc/resolv.conf` y comprobar que contiene un servidor de nombres válido.

A.2. Manual de uso de VNUML

Virtual Network User-Mode Linux o VNUML es una herramienta de virtualización de propósito general que permite trabajar con el software User-Mode Linux. VNUML permite traspasar la mayor parte de los complejos detalles de User-Mode Linux consiguiendo trabajar de una forma más fácil y rápida que utilizando las directivas User-Mode Linux de forma directa.

La herramienta VNUML está formada por dos componentes principales:

- El lenguaje XML utilizado para definir y configurar los escenarios. Mediante el uso correcto de este lenguaje XML podemos añadir nuevas máquinas virtuales a la simulación o indicar que una determinada máquina arranque un servidor en concreto.
- El intérprete de dicho lenguaje que nos permite ejecutar las simulaciones e interactuar con ellas.

Comprender el funcionamiento de una simulación es mucho más sencillo si se tiene una de ejemplo. A continuación se muestra el código XML necesario para crear una simulación con 10 VLANs. Se puede ver cómo hay una parte en la que se establece la versión de VNUMLPARSER que se va a utilizar y el nombre que se le da a la simulación. Del mismo modo, en la sección "vm_defaults" se establecen los valores por defecto para todas las máquinas virtuales que se van a utilizar en la simulación. Esta parte es muy importante ya que establece el kernel y el filesystem que se va a utilizar en la simulación. Al final del siguiente listado de código se puede ver cómo se declara una máquina virtual y en la parte en la que están los puntos suspensivos habría que escribir el código XML de todas y cada una de las máquinas virtuales que sea necesario incluir en la simulación.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE vnuml SYSTEM "/usr/local/share/xml/vnuml/vnuml.dtd">

<vnuml>
```

```
<!-- Declaraciones globales para la simulación -->
<global>
  <version>1.7</version>
  <simulation_name>6IX_96</simulation_name>
  <ssh_version>1</ssh_version>
  <ssh_key>/root/.ssh/identity.pub</ssh_key>
  <automac offset="0"/>
  <vm_mgmt type="private" network="192.168.1.0" mask="24" offset="44">
    <host_mapping />
  </vm_mgmt>
<!-- Valores por defecto para todas las máquinas de la simulación -->
  <vm_defaults>
    <filesystem type="cow">/usr/local/share/vnuml/filesystems/root_fs</filesystem>
    <kernel>/usr/local/share/vnuml/kernels/linux</kernel>
    <basedir>/root/simulacion_UCM/</basedir>
    <console id="0">xterm</console>
    <console id="1">pts</console>
  </vm_defaults>
</global>

<!-- Declaraciones de las redes que van a aparecer en la simulacion -->
<net name="VLAN1" mode="virtual_bridge" external="eth0" vlan="1"/>
<net name="VLAN2" mode="virtual_bridge" external="eth0" vlan="2"/>
<net name="VLAN3" mode="virtual_bridge" external="eth0" vlan="3"/>
<net name="VLAN4" mode="virtual_bridge" external="eth0" vlan="4"/>
<net name="VLAN5" mode="virtual_bridge" external="eth0" vlan="5"/>
<net name="VLAN6" mode="virtual_bridge" external="eth0" vlan="6"/>
<net name="VLAN7" mode="virtual_bridge" external="eth0" vlan="7"/>
<net name="VLAN8" mode="virtual_bridge" external="eth0" vlan="8"/>
<net name="VLAN9" mode="virtual_bridge" external="eth0" vlan="9"/>
<net name="VLAN10" mode="virtual_bridge" external="eth0" vlan="10"/>

<!-- Host de la vlan VLAN1 -->
<vm name="ucm1">
  <if id="1" net="VLAN1">
    <ipv4 mask="255.255.255.0">147.96.3.4</ipv4>
    <ipv6 mask="/96">2001:720:1500:4200:100:3:0:4</ipv6>
  </if>
  <route type="ipv4" gw="147.96.3.1">default</route>
  <route type="ipv6" gw="2001:720:1500:4200:100:3:0:1">default</route>
</vm>
...
...
...
</vnuml>
```

A continuación, a modo de ejemplo y sin entrar mucho en detalle sobre lo que hace cada parámetro en concreto, se muestran las líneas en XML que son necesarias para crear la máquina virtual que hará la función de servidor primario de nombres, es decir, servidor DNS primario.

```
<vm name="ucdns">
   <if id="1" net="Red3">
      <ipv4 mask="255.255.255.0">147.96.2.4</ipv4>
      <ipv6 mask="/96">2001:720:1500:4200:600:2:0:4</ipv6>
   </if>
   <route type="ipv4" gw="147.96.2.1">default</route>
   <route type="ipv6" gw="2001:720:1500:4200:600:2:0:1">default</route>
   <filetree root="/etc/bind/" when="copiar">/conf_bind/</filetree>
   <exec seq="servidores" type="verbatim">named</exec>
</vm>
```

Mediante estas líneas se dota a la máquina virtual de un nombre, en este caso "ucdns", dos direcciones IP, reglas de enrutado y unas reglas de comportamiento que explicaremos más adelante.

- Mediante el uso de <*vm*> y <*vm*> se define la máquina virtual y se le asigna un nombre.
- Mediante el uso de <*if*> y <*if*> se definen los distintos interfaces de red que tendrá la máquina virtual y se dota a cada uno de ellos de una dirección IP. Puede ser una dirección IPv4, IPv6, ambas o incluso ninguna y que le sea asignada mediante DHCP una vez que la simulación ha sido lanzada.
- Mediante el uso de <*route*> y <*route*> se define la tabla de rutas de la máquina virtual. En este ejemplo se indica la puerta de enlace predeterminada de nuestra máquina para las conexiones de IPv4 y de IPv6.
- Mediante el uso de <*filetree*>, <*filetree*>, <*exec*> y <*exec*> se dota a la máquina virtual de un comportamiento más personalizado acorde con las necesidades de la simulación. La etiqueta <*filetree*> se utiliza para copiar archivos desde el host anfitrión a la máquina virtual. En este ejemplo, se utiliza para copiar los archivos de configuración del servidor DNS a la máquina virtual. Por otro lado, la etiqueta <*exec*> se utiliza para ejecutar cualquier comando en la máquina virtual que decidamos. En este caso, se utiliza para ejecutar el servidor bind9 en "ucdns".

A continuación, se va a explicar los pasos necesarios para ejecutar la simulación con VNUML Parser. Para ello es imprescindible disponer de un ordenador con el Sistema Operativo GNU/Linux instalado, correctamente configurado en el que tengamos acceso a una cuenta de superusuario o *root* (debido a las características de la simulación, es necesario permisos de *root* para que se ejecute correctamente). Además es necesario diponer de la herramienta VNUMLPARSER, un sistema de ficheros y un kernel (su correcta instalación se ha explicado en el apartado anterior).

Pasos necesarios para desplegar la simulación:

1. Entrar en el directorio en el que se encuentra en archivo de con la simulación:

```
cd /root/simulacion_UCM/
```

2. Comprobación de que la configuración de la simulación es correcta y ejecución del escenario:

   ```
   vnumlparser -t complu6IX_96_gui.xml
   ```

3. Copiar los archivos de configuración de los distintos servidores y clientes.

   ```
   vnumlparser -x copiar@complu6IX_96_gui.xml
   ```

4. Ejecución de los servidores involucrados en la simulación (DNS primario, DNS secundario, correo, Web, FTP) cada uno en la máquina virtual que le corresponde:

   ```
   vnumlparser -x servidores@complu6IX_96_gui.xml
   ```

5. Realización de todas las pruebas sobre la simulación.

6. Finalización de la simulación y destrucción del escenario:

   ```
   vnumlparser -d complu6IX_96_gui.xml
   ```