

PROYECTO FINAL DE CARRERA



Escuela Técnica
Superior de Ingeniería
Informática

Instalación de una infraestructura de red IPv6

Autor: José Miguel Martínez Bier

Director: Carlos Miguel Tavares Calafate

Curso 2009/2010



Índice

1	Introducción.....	4
1.1	Objetivos del PFC.....	5
1.2	Estructura de la memoria.....	5
2	Descripción del protocolo IPv6	6
2.1	Ubicación en la pila TCP/IP.....	6
2.2	Formato de direcciones.....	7
2.3	Prefijos especiales	7
2.4	Autoconfiguración (configuración stateless).....	8
2.5	Detección de direcciones duplicadas.....	8
2.6	Formato de cabecera.....	9
2.7	Gestión de cabeceras opcionales (extension headers).....	10
2.8	Tráfico con prioridad.....	10
2.9	Características de seguridad.....	10
2.10	Direcciones unicast, anycast y multicast.....	11
2.11	Ámbitos de direcciones.....	11
2.12	Direcciones reservadas	13
3	Infraestructura.....	15
3.1	Preparación del entorno.....	15
4	Asignación de direcciones en la red.....	19
4.1	Métodos para obtener dirección IP en la red.....	19
4.2	Asignación de dirección para el router (pc1).....	19
4.3	Asignación de dirección para el servidor DNS (pc2).....	21
4.4	Asignación para clientes.....	22
4.4.1	Configuración stateless.....	22
4.4.2	Dirección IP establecida mediante un prefijo proporcionado por el router:.....	23
4.4.3	Dirección IP completamente asignada por el router (DHCPv6).....	24
4.5	Obtención de direcciones por parte de los clientes:	27
4.5.1	Cliente 1 (Sistema Debian).....	27
4.5.2	Cliente 2 (Sistema Windows Vista).....	31
5	Configuración y prueba de servicios.....	35
5.1	DNS.....	35
5.2	FTP.....	38
5.3	Web (HTTP).....	40
5.4	Vídeo en multicast.....	42
5.5	SSH/SCP.....	45
6	Conclusiones.....	47
7	Bibliografía.....	49

Índice de ilustraciones

Ilustración 1:	Hosts conectados a modem/router EDIMAX.....	15
Ilustración 2:	Desactivación de servidor DHCP en modem/router.....	16
Ilustración 3:	Clonar dirección MAC autorizada desde host a modem/router.....	17
Ilustración 4:	Obtención de dirección IPv6 mediante prefijo.....	23
Ilustración 5:	Activación exclusiva de IPv6.....	31
Ilustración 6:	Captura de tráfico del cliente DHCPv6 de Windows Vista.....	32

Ilustración 7: Configuración de red Windows Vista (shell).....	33
Ilustración 8: Configuración de red Windows Vista (GUI).....	33
Ilustración 9: Conexión por FTP desde Windows Vista (shell).....	37
Ilustración 10: Sesión de gFTP.....	38
Ilustración 11: Consulta de web desde Internet Explorer.....	39
Ilustración 12: Consulta de web especificando IP.....	40
Ilustración 13: Streaming desde el servidor con VLC.....	42
Ilustración 14: Vídeo en streaming desde cliente con VLC.....	43
Ilustración 15: Captura de tráfico durante sesión SSH.....	45

1 Introducción

IP es el protocolo de la capa de red de la pila TCP/IP, ampliamente utilizada por empresas e instituciones a nivel mundial, ya estén conectadas o no a la red Internet. La función del protocolo IP es, básicamente, asignar direcciones a hosts como identificación dentro de un grupo o red.

La versión 4 del protocolo IP, usada masivamente hoy en día por instituciones, empresas y particulares, empieza a presentar algunas deficiencias como la escasez de direcciones disponibles. La versión 6 resuelve este problema, además de aportar otras mejoras en seguridad y rendimiento, por lo que su adopción empieza a despertar cierto interés.

El RFC que describe esta versión (6) del protocolo IP data de 1998, aunque no se ha popularizado hasta tiempos recientes.

1.1 Objetivos del PFC

El objetivo del proyecto es implantar una pequeña infraestructura de red con soporte para la versión más reciente -a día de hoy- del protocolo IP (IP Versión 6).

La infraestructura de red implantada da soporte a la comunicación entre hosts, brindando servicios básicos bien conocidos (ICMP, FTP, HTTP, DNS, DHCP, SSH)

Se pretende comprobar la disponibilidad y soporte del protocolo IPv6 en sistemas operativos bien conocidos (familias Windows y GNU/Linux).

1.2 Estructura de la memoria

Las posteriores secciones de la memoria describen de la siguiente forma el proyecto realizado:

- Sección 2 (Descripción del protocolo IPv6): se describen los nuevos aspectos técnicos que implementa el protocolo, formato de las direcciones y cabeceras
- Sección 3 (Infraestructura y dispositivos utilizados): en esta sección figura el entorno donde se han realizado las pruebas (Laboratorio de Redes de la ETSINF) y cómo se ha debido preparar la infraestructura física antes de proceder a configurar el software de red en nodos cliente y nodos servidor.
- Sección 4 (Asignación de direcciones en la red): se detallan los distintos métodos disponibles en IPv6 para asignar identificadores a cada nodo, y cómo se ha procedido en cada uno de los cuatro hosts de la red.
- Sección 5 (Configuración y prueba de servicios): cómo se han instalado y configurado los distintos servicios básicos dentro de la red. Se incluyen capturas de tráfico de red y los resultados de conexiones y transferencias entre hosts.
- Sección 6 (Conclusiones): sección donde se recogen las impresiones obtenidas tras completar la instalación de la infraestructura de red, como el soporte por parte de aplicaciones y sistemas operativos o la posibilidad de migración de IPv4 a IPv6 por parte de usuarios y empresas.
- Sección 7: Bibliografía.

2 Descripción del protocolo IPv6

2.1 Ubicación en la pila TCP/IP

El modelo ISO/OSI describe una pila de 7 niveles para la comunicación entre hosts remotos.

Aplicación
Presentación
Sesión
Transporte
Red
Enlace
Física

En cambio, la pila TCP/IP es una simplificación con 5 niveles, que son los siguientes:

Aplicación
Transporte
Red
Enlace
Física

Tal y como está señalado en la tabla anterior, el protocolo IP se ubica en el nivel de Red. El propósito de la separación entre niveles, y es algo que se ha podido comprobar tras la implantación de la red, es aislar la funcionalidad y características de cada nivel, de forma que se puedan reutilizar o ampliar según convengan. Es decir, dentro de la capa de transporte, el protocolo TCP puede servir tanto para las aplicaciones HTTP o FTP, mientras que el protocolo UDP puede utilizarse por DNS o por otra aplicación escrita por el usuario.

Dado que las aplicaciones pueden reutilizar de forma transparente protocolos de capas inferiores, es lógico pensar que pueden reimplementarse éstos para ofrecer nuevas funcionalidades a las aplicaciones. Por ejemplo, la modificación de los protocolos IP y DNS dentro de una red permiten que un usuario/desarrollador continúe utilizando una página web de forma transparente, sin percibir que los protocolos de las capas inferiores han variado.

2.2 Formato de direcciones

Mientras que en IPv4 se usan 32 bits para representar cada dirección (de 4294967296 posibles), en IPv6 la longitud de este campo¹ es de 128 bits, lo que permite representar la cifra de 340282366920938463463374607431768211456 direcciones posibles; en términos más manejables, son $3,4 \cdot 10^{38}$ valores distintos, o, visto de otra forma, $5,67 \cdot 10^{28}$ por cada una de las personas de este planeta (asumiendo que son 6.000 millones, aproximadamente)

Los 128 bits se dividen en partes de 16 bits, y se separan por el delimitador :

De esta forma, una dirección puede ser `fedc:ba65:7654:2333:fedb:ba77:7655:3211`, y otra podría ser `1080:0:0:0:8:400:200C:417A`. Aunque algunos campos tengan valor 0, es necesario incluirlos en principio.

Sin embargo, cuando se repiten varios campos con valor 0, existe la posibilidad de abreviar la dirección utilizando otra notación, sustituyendo los campos nulos por el delimitador ::

Con este método, la dirección anteriormente indicada `1080:0:0:0:8:400:200C:417A`, pasaría a expresarse como `1080::8:400:200C:417A`. El delimitador especial :: sólo puede utilizarse una vez para expresar la repetición de varios campos nulos, con objeto de evitar ambigüedades.

A la hora de representar un prefijo que identifique un conjunto de direcciones, se declara una dirección seguida de su longitud. Partiendo de un prefijo como `12AB0000000CD3` (60 bits), su representación sería `12AB:0:0:CD30::/60`

Pueden existir entornos mixtos donde un nodo IPv6 pueda manejar en su propio formato direcciones IPv4². Por ejemplo, un nodo con dirección IPv4 `222.1.41.90`, puede ser identificado por otro nodo IPv6 con la dirección `::222.1.41.90`.

2.3 Prefijos especiales

`::1/128` Sin especificar

`::1/128` Dirección loopback (comunicación de un nodo consigo mismo)

`ff00::/8` Dirección multicast

`fe80::/10` Dirección unicast de ámbito link-local

`fec0::/10` Dirección unicast de ámbito site-local

Cualquier otro prefijo pertenece a una dirección de ámbito global. El concepto de ámbito se desarrolla más adelante.

¹ Internet Protocol, Version 6 (IPv6) Specification <http://www.faqs.org/rfcs/rfc2460.html>

² The TCP/IP Guide: IPv6/IPv4 Address Embedding

http://www.tcpiptide.com/free/t_IPv6IPv4AddressEmbedding-2.htm

2.4 Autoconfiguración (configuración stateless)

IPv6 incluye una característica que permite a los nodos formar por sí mismos una de sus direcciones IP (de hecho, pueden tener varias direcciones IP si tienen prefijos distintos) El propósito es poder integrarse sencilla y rápidamente en una red con una dirección propia. La técnica para obtener una dirección única es partir de un prefijo (usualmente el prefijo `fe80::/10` para direcciones de ámbito local, u otro distinto ofrecido por un router), y completarlo con una variación de la dirección física (dirección MAC).

La configuración stateless la aplican habitualmente los nodos tras levantar sus interfaces de red, lo que explica que dispongan de una dirección IPv6 aunque el usuario no haya configurado nada a tal efecto.

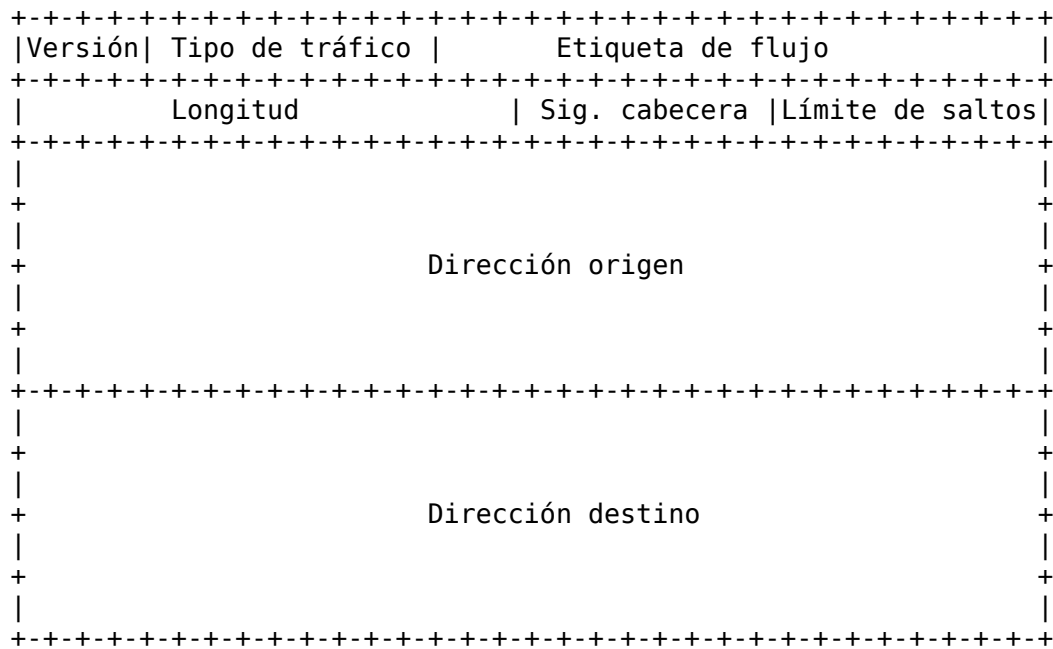
Por ejemplo, un nodo con dirección MAC `00-0C-6E-73-B0-0F`, puede autoconfigurarse en el arranque y establecer una dirección IP similar (`fe80::20c:6eff:fe73:b00f`) El mecanismo para obtener la dirección IP a partir de un prefijo y la dirección física de 48 bits se detallará en la sección de asignación de direcciones.

2.5 Detección de direcciones duplicadas

EL formato de direcciones de IPv6 asegura, en principio, que habrá direcciones disponibles para cualquier host que quiera obtener una. Sin embargo, pueden darse casos de colisiones en los que dos nodos compartan IP. Se proveen mecanismos para recuperar la conexión tan pronto como sea posible (Optimistic Duplicate Address Detection).

2.6 Formato de cabecera

Cada cabecera de un paquete IPv6 tiene el siguiente formato³:



Versión

4 bits para identificar la versión del protocolo IP (en este caso, valor 6)

Tipo de tráfico

Campo de 8 bits para indicar el tipo de tráfico. Este campo lo utilizan los nodos origen para que los routers lo inserten en la cola que consideren más oportuna.

Etiqueta de flujo

20 bits para indicar el flujo al que pertenece el paquete. Esto permite a los routers recordar el tratamiento que se ha dado a otros paquetes del flujo y aplicarlo más rápidamente

Longitud

16 bits que indican la longitud en bytes del resto del paquete

Siguiente cabecera

Campo de 8 bits que representa el tipo de la siguiente cabecera (puede haber varias)

3 The IPv6 Header and How it Works <http://ipv6.com/articles/general/IPv6-Header.htm>

Límite de saltos

Entero de 8 bits. Cada nodo que encamina el paquete decrementa en uno este valor. En caso de llegar a 0, se descarta.

Dirección origen

Dirección de 128 bits de quien originó el paquete.

Dirección destino

Dirección de 128 bits del destinatario del paquete.

Además, el tamaño predeterminado de MTU para paquetes IPv6 dentro de una red Ethernet es de 1500 bytes.

2.7 Gestión de cabeceras opcionales (extension headers)

Se mejora el tratamiento de las cabeceras, haciendo que las opciones viajen en cabeceras separadas que -habitualmente- no son examinadas por los routers que encaminan el paquete hasta llegar al destino. Las cabeceras opcionales se sitúan entre la principal y la cabecera del nivel superior. De este tipo de cabeceras pueden incluirse 0, 1 ó más.

2.8 Tráfico con prioridad

Como se adivinaba examinando el formato de las cabeceras IPv6, es posible asignar distintas prioridades a según qué paquetes, para obtener mejor rendimiento con algunas aplicaciones (por ejemplo, streaming de vídeo)

2.9 Características de seguridad

IPv6 obliga a dar soporte a IPsec. IPsec es un conjunto de estándares para definir políticas de seguridad en comunicaciones IP. Características que proporciona:

- Confidencialidad: el tráfico IPsec se cifra mediante los algoritmos DES y triple DES.
- Autenticación del origen: se incluye una suma de comprobación (checksum) que incorpora una clave compartida, con lo que el receptor tiene la certeza de que fue realmente enviado por el otro extremo.

- Integridad: la suma de comprobación también sirve para comprobar que el paquete no ha sido alterado durante su viaje desde el origen hasta el destino.

2.10 *Direcciones unicast, anycast y multicast*

En función del número de destinatarios que se asocian a una dirección IPv6, tenemos tres tipos:

- Dirección Unicast: es un identificador para una sola interfaz de red. El paquete que se envíe a una dirección unicast sólo será recibido por ese destinatario.
- Dirección Anycast: la dirección identifica a un conjunto de interfaces. Un paquete enviado a una dirección anycast se entrega a uno de los nodos asociados (el más cercano, atendiendo a la métrica de la red)
- Dirección Multicast: la dirección identifica a un conjunto de interfaces. La diferencia con las direcciones anycast es que los paquetes se entregan a todos los nodos asociados con la dirección multicast.

En IPv6 no hay direcciones broadcast para difusión. Para ese cometido, se utilizan las direcciones multicast.

2.11 *Ámbitos de direcciones*

Cada dirección IP tiene un ámbito, es decir, un área donde el identificador de una interfaz (o conjunto de interfaces) tiene validez. Los routers reconocen el ámbito de una dirección a partir de su prefijo y deciden si encaminar su tráfico o no. Existen los siguientes ámbitos⁴:

Prefijo	Tipo de dirección
::/128	Sin especificar
::1/128	Loopback
FF00::/8	Multicast
FE80::/10	Link-local unicast
FEC0::/10	Site-local unicast
Otros	Global unicast

Es decir, un nodo no puede solicitar contactar con una IP con prefijo `fe80::/10` que no esté en su red (conectado al mismo enlace), pues requiere encaminamiento por parte de un router y denegará la operación por salir del ámbito local. Sin embargo, una conexión con una IP como `2a00:1450:8006::68` no debería denegarse por cuestión de ámbito, pues se encuadra en un ámbito global.

⁴ IPv6 Addressing Architecture (RFC 3513) <http://www.ietf.org/rfc/rfc3513.txt>

2.12 Direcciones reservadas

Existe un grupo de direcciones multicast reservadas que nunca deberán utilizarse para identificar a un grupo multicast:

FF02::1	FF02::2	FF02::3	FF02::4
FF02::5	FF02::6	FF02::7	FF02::8
FF02::9	FF02::A	FF02::B	FF02::C
FF02::C	FF02::D	FF02::E	FF02::F
FF02::12	FF02::16	FF02::6A	FF02::6B
FF02::6C	FF02::6D	FF02::6E	FF02::6F
FF02::FB	FF02::1:1	FF02::1:2	FF02::1:3
FF02::1:4			
FF02::1:FF00:0000/104			
FF02:0:0:0:0:2:FF00::/104			

Son direcciones especiales cuyo propósito ya está definido⁵ por la Internet Assigned Numbers Authority (IANA); por ejemplo, FF02::1 se utiliza para enviar peticiones a todos los nodos, FF02::2 a todos los routers y FF02::1:2 a todos los agentes DHCP.

Las dos siguientes direcciones (All Nodes Addresses) identifican a todos los nodos IPv6 en los ámbitos node-local o link-local (FF01::1, FF02::1)

Para identificar a todos los routers IPv6 dentro de los ámbitos node-local, link-local o site-local, se utilizan respectivamente las direcciones FF01::2, FF02::2 y FF05::2

⁵ IPv6 Multicast Address Space Registry
<http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xml>

3 Infraestructura

Se implanta la red local en el Laboratorio de Redes de la ETSINF (edificio de la antigua EUI) de la Universidad Politécnica de Valencia. Se dispone de tres ordenadores con la siguiente configuración:

- CPU AMD Athlon XP 2500+
- Disco duro Seagate ST3250820A ATA de 232GB
- 1GB de memoria RAM
- Tarjeta de red integrada VIA VT6102.

Además, se añade un cuarto host a la red, un portátil Samsung R519, del que aprovecharemos la partición con Windows Vista preinstalado, para que actúe como cliente, con las siguientes características:

- CPU Intel Dual Core 2.4GHz
- 4GB de de memoria RAM
- Disco duro ST9250315AS ATA de 250GB.
- Tarjeta de red Realtek RTL8101E.

Los cuatro ordenadores se interconectan mediante cableado Ethernet a un modem/router modelo Edimax BR-6104KP , quedando la LAN aislada, en principio, de la red del laboratorio.



Ilustración 1: Hosts conectados a modem/router EDIMAX

3.1 Preparación del entorno

Por defecto, el modem/router lleva habilitado un servidor DHCP (para IP versión 4), con objeto de facilitar el acceso a Internet a varios ordenadores en entornos domésticos. La IP versión 4 del modem/router (necesaria para acceder a los menús de configuración) es 192.168.1.1, y el par usuario/contraseña establecido por defecto es admin/1234.

La configuración se realiza accediendo vía web a la IP 192.168.1.1 e introduciendo el par anteriormente descrito.

El hecho de que el servidor DHCP para IP versión 4 venga activado por defecto, es un inconveniente para conseguir el correcto funcionamiento de la red local. En primer lugar, no deseamos que el modem/router asigne a cada cliente la configuración del servidor DNS, pues este servicio lo vamos a implantar en uno de nuestros hosts. En segundo lugar, la configuración ofrecida por el modem/router es para IP versión 4, y en nuestra red pretendemos comprobar que se puede operar exclusivamente con la versión 6.

Sin embargo, cabe destacar que el dispositivo entrega sin problemas los paquetes de IPv6 dentro de la red, aunque su funcionamiento y configuración estén orientados principalmente

para operar con la versión 4.

Así pues, accedemos a la configuración del modem/router para desactivar el servidor DHCP, puesto que el direccionamiento IP de la red lo estableceremos nosotros, ya sea de forma estática o mediante DHCPv6.

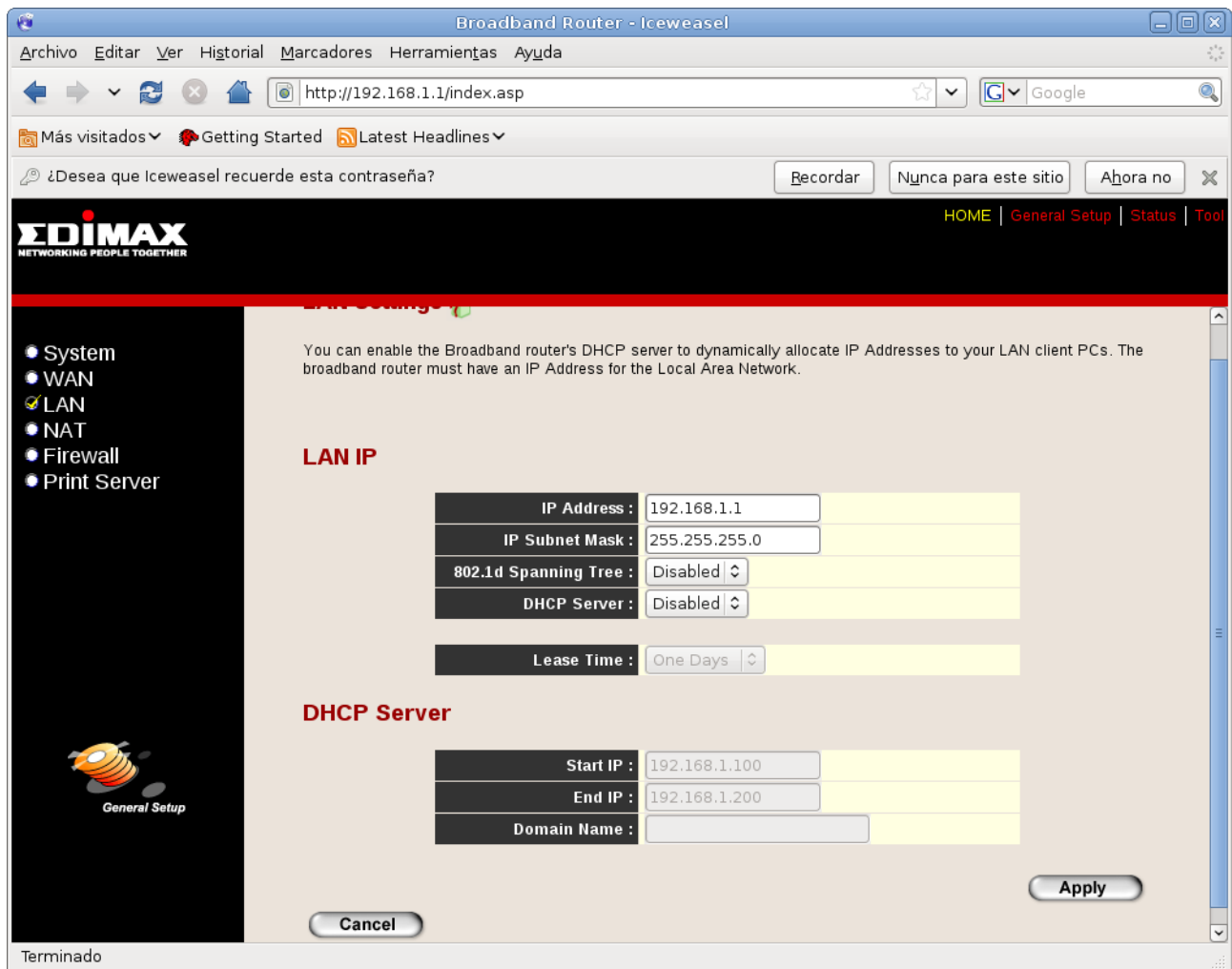


Ilustración 2: Desactivación de servidor DHCP en modem/router

En algún momento podemos querer conectar el modem/router a la red del laboratorio para tener acceso a Internet. Para ello, conectaríamos un cable Ethernet desde la salida WAN del modem/router hasta una de las tomas del laboratorio. Un inconveniente añadido es que la dirección MAC del modem/router no está en la lista de acceso del centro de cálculo (para impedir que se conecten ordenadores ajenos a la red de la universidad sin pasar por los debidos controles). Conectar directamente el modem/router haría saltar alertas de tipo SNMP en el centro de cálculo, con lo que necesitamos que opere con una dirección MAC autorizada. Para ello, el dispositivo permite actualizar su dirección MAC copiándola de otro host conectado al mismo.

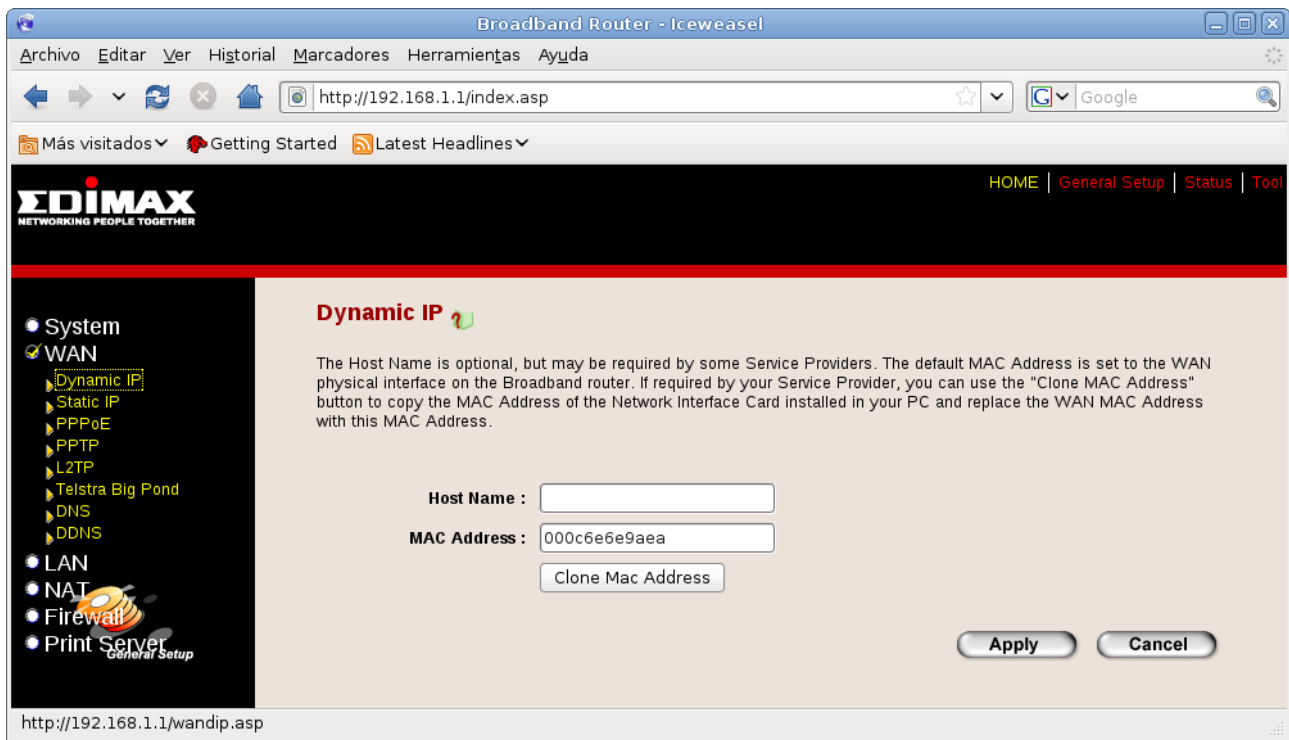


Ilustración 3: Clonar dirección MAC autorizada desde host a modem/router

Con esto, el dispositivo graba su nueva dirección MAC (se puede recuperar la original con el procedimiento de Reset, pulsando el botón al efecto en la parte trasera) y queda operativo dentro de la red de la UPV. Sin embargo, y este es uno de los factores que más han afectado a la implantación de la red, la infraestructura de la UPV no permite por el momento el acceso a sus recursos de red mediante IP versión 6. Con ello, tener el modem/router con una dirección MAC válida permite a los ordenadores conectados al mismo enviar tráfico a través de la red de la universidad, con la salvedad de que debe ser dentro de paquetes IP versión 4.

4 Asignación de direcciones en la red

Una vez que un nodo se conecta a una red, necesita una dirección, que no es más que un identificador que utilizará la capa de red para anunciarse (distinguirse) entre el resto de nodos vecinos de la red. La obtención de este parámetro es imprescindible para empezar a operar en el entorno de red y utilizar los distintos servicios implantados.

4.1 Métodos para obtener dirección IP en la red

En IPv6, un nodo puede conocer/obtener su dirección de varias formas:

- dirección IP asignada de forma estática. En nuestro caso, se ha optado por asignar la dirección de forma fija a los nodos que desempeñarán funciones imprescindibles en la red. Como ejemplo, el servidor DNS utiliza la dirección `fc00::1001`, que debe ser bien conocida por el resto de clientes/servidores de la red.
- Dirección de ámbito local establecida unilateralmente por el cliente; el nodo se autoconfigura utilizando un prefijo del tipo `fe80` y completa la dirección aplicando una función sobre su propia dirección MAC (dirección hardware o dirección de la capa de enlace)
- Dirección IP establecida mediante un prefijo proporcionado por el router. En este caso, el router proporciona parte de la dirección (el prefijo) y el nodo que desea obtener una dirección se encarga de completarla.
- Dirección IP completamente asignada por el router. Con objeto de controlar completamente qué direcciones utilizan los miembros de una red, es posible ofrecer a estos la dirección íntegra mediante el protocolo DHCPv6 (aunque exista la posibilidad de utilizar la configuración stateless, la asignación puede establecerse de forma similar a DHCP para IPv4)

En resumen, en la red se han asignado las direcciones de la siguiente forma:

pc1	pc2	cliente1	cliente2
estática	estática	DHCPv6	DHCPv6

Cabe aclarar que, en los hosts de la red que utilizan GNU/Linux, se han desactivado applets del tipo Network Manager, por interferir en las tareas de asignación de direcciones IP.

4.2 Asignación de dirección para el router (pc1)

Para especificar en el pc1 (sistema Debian) que la dirección es estática, se debe editar el fichero `/etc/network/interfaces` para que contenga las siguientes líneas:

```
iface eth0 inet6 static
address fc00::1000
netmask 64
```

```
iface eth0 inet static
address 192.168.1.101
netmask 255.255.255.0
gateway 192.168.1.1
```

Con ello, se especifica que la interfaz `eth0` tendrá una dirección IP estática, configurada como `fc00::1000` (versión 6).

Además, para poder acceder vía web al modem/router, necesitamos especificar que la interfaz `eth0` también tendrá una dirección IPv4, tal y como figura en el segundo bloque del fichero `/etc/network/interfaces`. Como se puede observar, pueden funcionar ambas versiones (4 y 6) de la pila TCP/IP en un mismo host; de hecho, en la interfaz loopback (`lo`) se utilizan las direcciones `127.0.0.1` y `::1`.

Una vez modificado el fichero de interfaces, tomará efecto en el próximo reinicio o con la orden

```
# /etc/init.d/networking restart
```

Tras terminar este proceso, se consulta el estado de las interfaces de red con la orden `ifconfig`; el resultado es:

```
# ifconfig

eth0      Link encap:Ethernet  HWaddr 00:0c:6e:6e:9a:ea
          inet addr:192.168.1.101  Bcast:192.168.1.255
Mask:255.255.255.0
          inet6 addr: fc00::1000/64 Scope:Global
          inet6 addr: fe80::20c:6eff:fe6e:9aea/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:867 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1476 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:94094 (91.8 KiB)  TX bytes:165184 (161.3 KiB)
          Interrupt:23 Base address:0xa000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
```

```
RX packets:176 errors:0 dropped:0 overruns:0 frame:0
TX packets:176 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:14864 (14.5 KiB) TX bytes:14864 (14.5 KiB)
```

4.3 Asignación de dirección para el servidor DNS (pc2)

Análogamente a lo que hicimos con el router (pc1), se edita el mismo fichero interfaces (también en un sistema Debian) para que contenga lo siguiente:

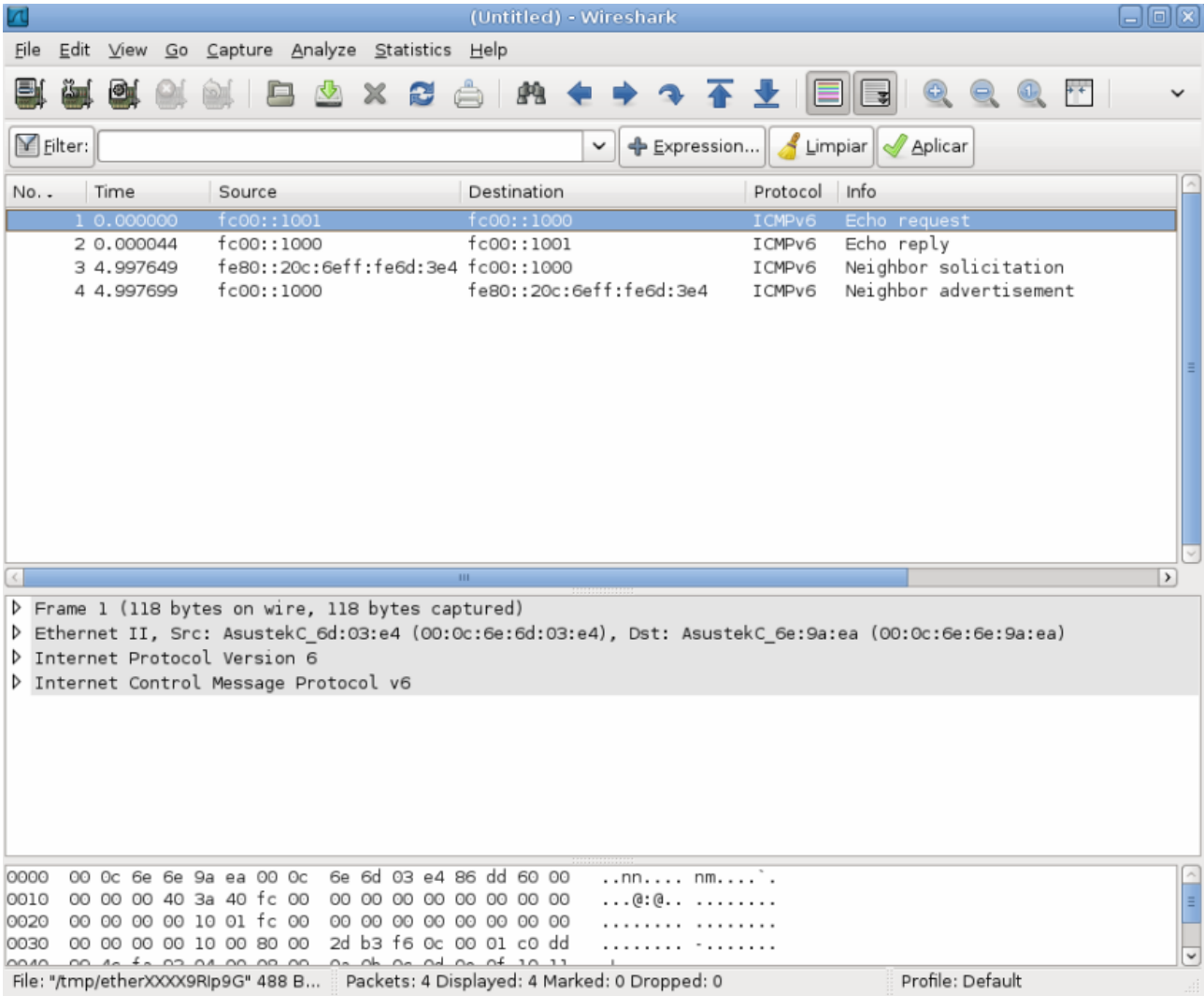
```
iface eth0 inet6 static
address fc00::1001
netmask 64
```

Nótese que para este host sólo especificamos dirección de versión 6 para la interfaz eth0, y para la interfaz lo tendrá ambas versiones igualmente (127.0.0.1 y ::1).

Una vez establecido el direccionamiento básico de los nodos clave en la infraestructura de red, es factible hacer una prueba de comunicación entre ellos, mediante un ping (ICMPv6). La orden utilizada desde el pc2 es:

```
~ ping6 -I eth0 -c 1 fc00::1000
```

Como se puede apreciar, el nodo con dirección `fc00::1001` solicita respuesta del `fc00::1000` mediante el protocolo ICMP versión 6. Los dos últimos eventos de la captura muestran un diálogo mediante el cual el pc2 intenta encontrar un router entre sus vecinos.



4.4 Asignación para clientes

4.4.1 Dirección IP establecida mediante un prefijo proporcionado por el router:

Para los nodos de la red que actúen como clientes, una opción para que estos obtengan su dirección es ofrecerles un prefijo que les indique la red a la que pertenecen y que completen la dirección con una derivación de su propia dirección MAC (método detallado en una sección anterior de la memoria). Para ello, es necesario establecer que el pc1 reenvíe los paquetes IPv6 que reciba; esto es, que realmente actúe como un router.

En un sistema Debian, este parámetro se establece activando la directiva `net.ipv6.conf.all.forwarding=1` en el fichero `/etc/sysctl.conf`. En principio, debido a un bug, esta directiva no se puede aplicar en el arranque del sistema escogido (Debian estable) por un conflicto de dependencias. La solución es, una vez el sistema ha arrancado completamente, cargar la configuración manualmente con la orden:

```
# sysctl -p /etc/sysctl.conf
```

Se requiere que este nodo actúe como router para poder anunciar prefijos desde el mismo. Para ello, nos serviremos del programa `radvd` (Router Advertisement Daemon) para informar periódicamente de que existe un router y ofrece prefijos con los que formar direcciones IP. Se instala con la orden:

```
# aptitude install radvd
```

El fichero de configuración que debemos editar es `/etc/radvd.conf`, que dejamos finalmente con el siguiente contenido:

```
interface eth0
{
    AdvSendAdvert on;
#   AdvManagedFlag on;
#   AdvOtherConfigFlag on;
    prefix 2001:db8::/64
    {
#   AdvAutonomous off;
    };
};
```

Realmente, el fichero descrito es el utilizado para el siguiente método de asignación (DHCPv6), con la salvedad de que, comentando esos 3 parámetros, sirve para ofrecer prefijos a los clientes. Así pues, las dos líneas necesarias para ofrecer prefijos a los clientes son:

```
AdvSendAdvert on;
prefix 2001:db8::/64
```

que indican, respectivamente, que el router anunciará prefijos y responderá a solicitudes de los mismos, y el contenido del propio prefijo ofrecido. El resto de parámetros se establecen para un correcto funcionamiento del DHCPv6 y se detallarán más adelante.

Una vez configurados los parámetros pertinentes, se puede lanzar el Router Advertisement Daemon con la orden:

```
# /etc/init.d/radvd restart
```

No. .	Time	Source	Destination	Protocol	Info
1	0.000000	::	ff02::1:ff34:4015	ICMPv6	Neighbor solicitation
2	0.000045	fe80::f4ed:ef71:ba34:4015	ff02::2	ICMPv6	Router solicitation
3	0.000067	fe80::f4ed:ef71:ba34:4015	ff02::16	ICMPv6	Multicast Listener Report Message v2
4	0.000467	fe80::20c:6eff:fe6e:9aea	ff02::1	ICMPv6	Router advertisement
5	0.000913	fe80::f4ed:ef71:ba34:4015	ff02::16	ICMPv6	Multicast Listener Report Message v2
6	0.024412	fe80::f4ed:ef71:ba34:4015	ff02::16	ICMPv6	Multicast Listener Report Message v2
7	0.032643	fe80::f4ed:ef71:ba34:4015	ff02::1:3	UDP	Source port: 53191 Destination port: llmnr
8	0.049077	fe80::f4ed:ef71:ba34:4015	ff02::1:ff6e:9aea	ICMPv6	Neighbor solicitation
9	0.049091	fe80::20c:6eff:fe6e:9aea	fe80::f4ed:ef71:ba34:4015	ICMPv6	Neighbor advertisement
10	0.054879	2001:db8::c5db:3670:8c70:d1e	fec0:0:0:ffff::3	DNS	Standard query AAAA time.windows.com
11	0.139754	fe80::f4ed:ef71:ba34:4015	ff02::1:3	UDP	Source port: 53191 Destination port: llmnr
12	0.512054	::	ff02::1:ff34:4015	ICMPv6	Neighbor solicitation

Ilustración 4: Obtención de dirección IPv6 mediante prefijo

Así, comprobamos que el router ofrece prefijos y el cliente consigue configurar una dirección con el prefijo 2001:db8 (evento número 10 de la captura)

4.4.2 Dirección de ámbito local establecida unilateralmente por el cliente (stateless autoconfiguración)

En este caso, el sistema arranca con una dirección de ámbito local (recordemos que pueden utilizarse varias direcciones con distintos prefijos) sin necesidad de especificar cuál es. En la red local implantada, los distintos sistemas arrancan con las siguientes direcciones en formato IEEE-EUI64 modificado:

pc1 (router) → fe80::20c:6eff:fe6e:9aea

pc2 (servidor DNS) → fe80::20c:6eff:fe6d:3e4

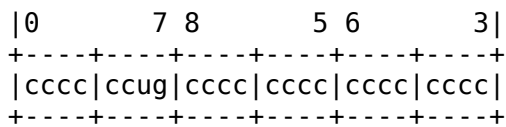
Cliente1 Debian → fe80::20c:6eff:fe73:b00f

Cliente2 Windows Vista → fe80::f4ed:ef71:ba34:4015

Las direcciones IEEE-EUI64 se forman a partir del prefijo fe80, y se completan con una derivación de la función MAC → 24 bits del fabricante + valor FFFE + 24 bits del producto. En la versión modificada⁶ se invierte el bit U/L (indica ámbito universal o local), que

⁶ IPv6 Interface Identifiers <http://msdn.microsoft.com/en-us/library/aa915616.aspx>

corresponde al bit número 6 de los primeros 3 bytes de la dirección MAC.



4.4.3 Dirección IP completamente asignada por el router (DHCPv6)

Se considera que el protocolo DHCPv6 configura las direcciones IP de los clientes en modo stateful. Es decir, de forma controlada por el servidor, a diferencia del modo stateless que pueden usar los clientes para formar su dirección a partir de un prefijo.

Los clientes usarán el puerto 546/UDP para el diálogo con los servidores, así como estos utilizarán el puerto 547/UDP.

Como hemos dicho anteriormente, las máquinas que prestan servicios en la red local se dejan configuradas con direcciones IP estáticas. El resto de clientes que se conectan a la red obtienen su dirección mediante DHCPv6. En realidad, la información que proporcionará el servidor DHCPv6 a los clientes no será exclusivamente la dirección IP, sino un conjunto de parámetros que facilitarán la integración de los mismos en la red, para que el usuario acceda a los recursos de una forma lo más transparente posible. Los nuevos clientes que se conecten a la red pueden solicitar mediante DHCPv6 información como:

- direcciones IP de los servidores que atenderán peticiones para resolver consultas de Servicio de Nombres de Dominio (DNS)
- prefijo para construir su dirección IP a partir del mismo.
- Dirección IP temporal
- Dirección IP preferida, por si el cliente no quiere que se le asigne otra distinta.
- Tiempo de renovación (tiempo tras el cual se reanudará la negociación para obtener la configuración de red)
- Nombre de la red

Para que el router tenga completo control de las direcciones que proporciona a los clientes (rango de direcciones y prefijo anunciado), es necesario establecer algunos parámetros para el Router Advertisement Daemon para que los clientes no completen su dirección a partir del prefijo ofrecido. Fichero `/etc/radvd.conf`

```
interface eth0
{
```



```

    AdvSendAdvert on;
    AdvManagedFlag on;
    AdvOtherConfigFlag on;
    prefix 2001:db8::/64
  {
    AdvAutonomous off;
  };
};

```

`AdvManagedFlag on;` # el servidor indica que los clientes configurarán su dirección de modo stateful (administrado por el servidor)

`AdvOtherConfigFlag on;` # Igual que el parámetro anterior, para el resto de campos que no sean la dirección IP.

`AdvAutonomous off;` # Indica que el prefijo especificado no está pensado para formar la dirección IP a partir del mismo.

Una vez modificado el comportamiento del Router Advertisement Daemon, se puede proceder a instalar el servidor DHCP para el sistema Debian:

```
# aptitude install wide-dhcpv6-server
```

El fichero de configuración para este servidor se encuentra en `/etc/wide-dhcpv6-server/dhcp6s.conf` y se puede dejar como sigue:

```

option domain-name-servers fc00::1001;
option domain-name "laboratorio.ln6";

host kame {
    duid 00:01:00:01:aa:bb;
    prefix 2001:db9:1111::/48 infinity;
};

```

```

# The followings are a sample configuration to provide an
IPv6 address
# from an address pool 2001:db8:1:2::1000-2000 for 3600[s].
# Note. You have to send an RA to fxp0; otherwise a client
cannot be sure
# about the prefix-length and the default router. If you
want to prevent
# stateless address configuration via RA, please set the
autonomous-flag to

```

```
# OFF in your RA configuration.

interface eth0 {
    address-pool pool1 3600;
};

pool pool1 {
    range fc00::6000 to fc00::8000 ;
};
```

Con las dos primeras líneas, se especificará a los clientes que deben utilizar `fc00::1001` como servidor DNS, y el dominio de la red será `laboratorio.ln6`.

El bloque encabezado por `host kame` es opcional; se deja como información de las capacidades del servidor. Significa que al host que presente ese DUID (identificador único de DHCP) se le servirá el prefijo indicado.

Los dos últimos bloques indican, respectivamente, que la información ofrecida tendrá una validez de 3600 segundos y el rango de direcciones que se ofrecerá a los clientes abarcará desde `fc00::6000` a `fc00::8000`.

4.5 Obtención de direcciones por parte de los clientes:

4.5.1 Cliente 1 (Sistema Debian)

Debemos instalar en este sistema la versión cliente de la aplicación `wide-dhcpv6`:

```
# aptitude install wide-dhcpv6-client
```

El fichero de configuración a modificar es `/etc/wide-dhcpv6/dhcp6c.conf`

```
# Default dhpc6c configuration: it assumes the address is
autoconfigured using
# router advertisements.
```

```
interface eth0
{
# information-only;

request domain-name-servers;
request domain-name;
send rapid-commit;
send ia-na 15;
```

```

    script "/etc/wide-dhcpv6/dhcp6c-script";
};

id-assoc na 15 { #

};

```

Como se puede observar en el comentario de la configuración por defecto, está pensado para obtener la dirección a partir de un prefijo determinado. En todo caso, las modificaciones oportunas en el servidor (indicadas anteriormente), permitirán que éste especifique exactamente todos los campos de la dirección IP.

Se deshabilita la opción `information-only`, que se usaría si el cliente no desea que se le asigne dirección desde el servidor. Se especifica que el cliente solicite servidores DNS (puede ser uno o más) y el nombre del dominio al que se une.

La opción `rapid-commit` acelera el intercambio de información en la transacción DHCP, simplificando el diálogo entre cliente y servidor. Está pensada para entornos de alta movilidad.

El parámetro IA-NA se establece para distinguir el diálogo DHCP entre cliente y servidor de los demás que se puedan dar en un determinado instante.

Finalmente se debe configurar la interfaz de red (eth0) para que obtenga su configuración de un servidor DHCPv6. Se modifica el fichero `/etc/network/interfaces` para que contenga:

```

# This file describes the network interfaces available on your
system
# and how to activate them. For more information, see
interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
#allow-hotplug eth0
auto eth0

iface eth0 inet dhcp
    up /etc/wide-dhcpv6/dhcp6c-ifupdown start
    down /etc/wide-dhcpv6/dhcp6c-ifupdown stop

```

Llegados a este punto, reiniciamos las interfaces de red:

```

# /etc/init.d/networking restart

```



```

eth0      Link encap:Ethernet  HWaddr 00:0c:6e:73:b0:0f
          inet6 addr: fc00::6001/128 Scope:Global
          inet6 addr: fe80::20c:6eff:fe73:b00f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:39 errors:0 dropped:0 overruns:0 frame:0
          TX packets:66 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4566 (4.4 KiB)  TX bytes:10159 (9.9 KiB)
          Interrupt:23 Base address:0xa000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:560 (560.0 B)  TX bytes:560 (560.0 B)

```

Examinando el contenido de `/etc/resolv.conf`, también vemos que se ha actualizado con el servidor DNS que especifica el servidor:

```

search laboratorio.ln6.
nameserver fc00::1001
search laboratorio.ln6.
nameserver fc00::1001

```

El script de actualización de `/etc/resolv.conf` puede funcionar incorrectamente en algunos casos, como se puede observar. Dado que habitualmente se acepta más de un servidor DNS por si el primero fallase, esto no llega a ser problemático y el cliente queda operativo dentro de la red.

También observamos un cambio en la tabla de encaminamiento del cliente. Se puede comparar si observamos la tabla en el momento de estar conectado por IPv4 a la red de la UPV (antes del diálogo con protocolo DHCPv6):

```

hostpfc3:~# route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
158.42.180.0    0.0.0.0         255.255.254.0  U        0      0      0 eth0
0.0.0.0         158.42.181.250  0.0.0.0        UG       0      0      0 eth0
hostpfc3:~# route -6n
Kernel IPv6 routing table
Destination      Next Hop        Flag Met Ref Use If
fe80::/64       ::              U    256 0   0 eth0
::/0            ::              !n  -1  1  164 lo
::1/128         ::              Un   0   1   37 lo
fe80::20c:6eff:fe73:b00f/128
ff00::/8        ::              U    256 0   0 eth0
::/0            ::              !n  -1  1  164 lo

```

Vemos que para IPv4 se encamina el tráfico a través de la pasarela `158.42.181.250`. Para el tráfico IPv6, se entrega todo de forma directa porque el nodo sólo maneja

direcciones propias y no conoce entre sus vecinos a ningún router IPv6.

Tras el intercambio de mensajes por DHCPv6, las tablas de routing quedan así:

```
hostpfc3:~# route -6n
Kernel IPv6 routing table
Destination          Next Hop              Flag Met Ref Use If
2001:db8::/64        ::                   UAe  256 0   1 eth0
fc00::6001/128       ::                   U   256 0   0 eth0
fe80::/64            ::                   U   256 0   0 eth0
::/0                  fe80::20c:6eff:fe6e:9aea UGDAe 1024 0 3 eth0
::/0                  ::                   !n   -1 1  20 lo
::1/128               ::                   Un   0  1   2 lo
fc00::6001/128       ::                   Un   0  1   9 lo
fe80::20c:6eff:fe73:b00f/128 ::                   Un   0  1  19 lo
ff00::/8              ::                   U   256 0   0 eth0
::/0                  ::                   !n   -1 1  20 lo

hostpfc3:~# route -n
Kernel IP routing table
Destination          Gateway              Genmask              Flags Metric Ref Use Iface
```

La tabla para IPv4 ha quedado vacía (no maneja ese tipo de direcciones) y para direcciones IPv6 que no sean propias, encamina el tráfico (campo next hop) hacia la dirección de ámbito local del router.

4.5.2 Cliente 2 (Sistema Windows Vista)

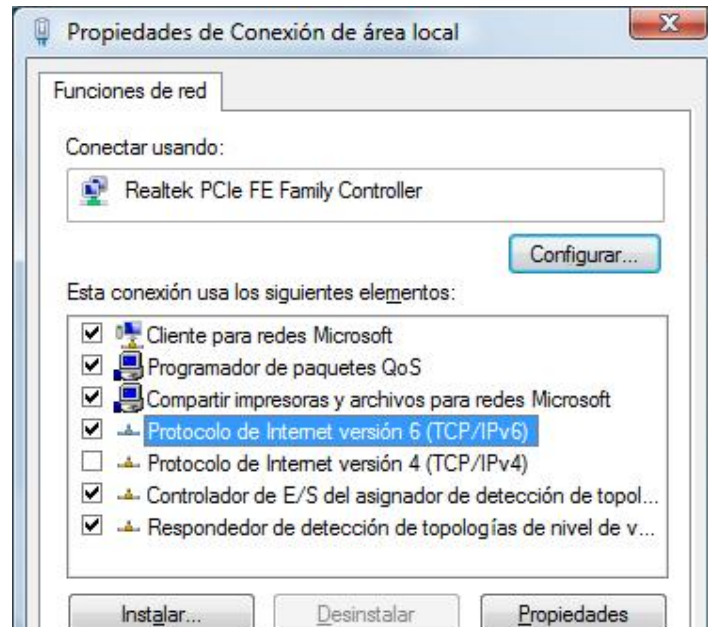
Uno de los clientes de la red objeto de estudio tiene como sistema operativo Windows Vista. Soporta de forma nativa IPv6 en su pila TCP/IP. Entre otros, Windows 7 también incluye el debido soporte para este protocolo y Windows XP es capaz de operar con él desde la salida del Service Pack 1 y posteriores.⁷

El cliente DHCP para IPv6 de Windows Vista puede operar en ambos modos Stateless (con prefijo anunciado por Router) y Stateful. En este sentido, no fue necesaria ninguna configuración adicional en el cliente, más que la propia obtención automática de IP y datos de la red. Funcionan ambos métodos, de forma que el sistema operativo opta por uno o por otro en función de la información que reciba desde el servidor.

Cabe decir que la integración de este cliente en la red se consiguió deshabilitando por completo la versión 4 de IP de la pila TCP/IP.

⁷Microsoft IPv6 <http://technet.microsoft.com/en-us/network/bb530961.aspx>

Ilustración 5: Activación exclusiva de IPv6



Ya sea reiniciando el PC cliente o la propia conexión de red, se registra un diálogo como el que se observa en la figura. El cliente solicita la presencia de un router entre sus vecinos y cuando recibe respuesta, comienza el diálogo para obtener la dirección IP, servidor DNS y el dominio de la red. Se envía al cliente:

- Dirección IP : fc00::6000
- Servidor DNS: fc00::1001
- Dominio: laboratorio.ln6

(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: + Expression... Limpia Aplicar

No.	Time	Source	Destination	Protocol	Info
1	0.000000	::	ff02::1:ff34:4015	ICMPv6	Neighbor solicitation
2	0.000076	fe80::f4ed:ef71:ba34:4015	ff02::2	ICMPv6	Router solicitation
3	0.000131	fe80::f4ed:ef71:ba34:4015	ff02::16	ICMPv6	Multicast Listener Report Message v2
4	0.000521	fe80::20c:6eff:fe6e:9aea	ff02::1	ICMPv6	Router advertisement
5	0.021074	fe80::f4ed:ef71:ba34:4015	ff02::1:2	DHCPv6	Solicit[Malformed Packet]
6	0.021270	fe80::20c:6eff:fe6e:9aea	fe80::f4ed:ef71:ba34:4015	DHCPv6	Advertise
7	0.041043	fe80::f4ed:ef71:ba34:4015	ff02::1:2	DHCPv6	Solicit[Malformed Packet]
8	0.041137	fe80::20c:6eff:fe6e:9aea	fe80::f4ed:ef71:ba34:4015	DHCPv6	Advertise
9	0.499955	fe80::f4ed:ef71:ba34:4015	ff02::16	ICMPv6	Multicast Listener Report Message v2
10	1.015475	fe80::f4ed:ef71:ba34:4015	ff02::16	ICMPv6	Multicast Listener Report Message v2
11	1.026774	fe80::f4ed:ef71:ba34:4015	ff02::1:ff6e:9aea	ICMPv6	Neighbor solicitation
12	1.026821	fe80::20c:6eff:fe6e:9aea	fe80::f4ed:ef71:ba34:4015	ICMPv6	Neighbor advertisement
13	1.041154	fe80::f4ed:ef71:ba34:4015	ff02::1:2	DHCPv6	Request[Malformed Packet]
14	1.041373	fe80::20c:6eff:fe6e:9aea	fe80::f4ed:ef71:ba34:4015	DHCPv6	Reply
15	1.042393	fe80::f4ed:ef71:ba34:4015	ff02::16	ICMPv6	Multicast Listener Report Message v2
16	1.059497	fc00::6000	fc00::1001	DNS	Standard query AAAA time.windows.com

Frame 14 (187 bytes on wire, 187 bytes captured)
 Ethernet II, Src: AsustekC_6e:9a:ea (00:0c:6e:6e:9a:ea), Dst: 00:24:54:13:c2:e0 (00:24:54:13:c2:e0)
 Internet Protocol Version 6
 User Datagram Protocol, Src Port: 53764 (53764), Dst Port: dhcpv6-client (546)
 DHCPv6
 Message type: Reply (7)
 Transaction-ID: 0x00045466
 Client Identifier
 Server Identifier
 Identity Association for Non-temporary Address
 DNS recursive name server
 option type: 23
 option length: 16
 DNS servers address: fc00::1001
 Domain Search List
 option type: 24
 option length: 17
 DNS Domain Search List
 Domain: laboratorio.ln6

Ilustración 6: Captura de tráfico del cliente DHCPv6 de Windows Vista

Se aprecia que Wireshark identifica los paquetes enviados desde el cliente Windows como incorrectos, aunque finalmente la operación se completa con éxito. De hecho, la consulta DNS que se registra al final de la captura ya tiene como origen la dirección `fc00::6000`, lo que indica que ha conseguido configurarse correctamente, al menos en lo que respecta a ese campo. Mediante la orden `ipconfig` consultamos que, efectivamente, toma la dirección IP y la puerta de enlace:


```
Administrador: C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. Reservados todos los derechos.

C:\Users\josemiguel>ipconfig

Configuración IP de Windows

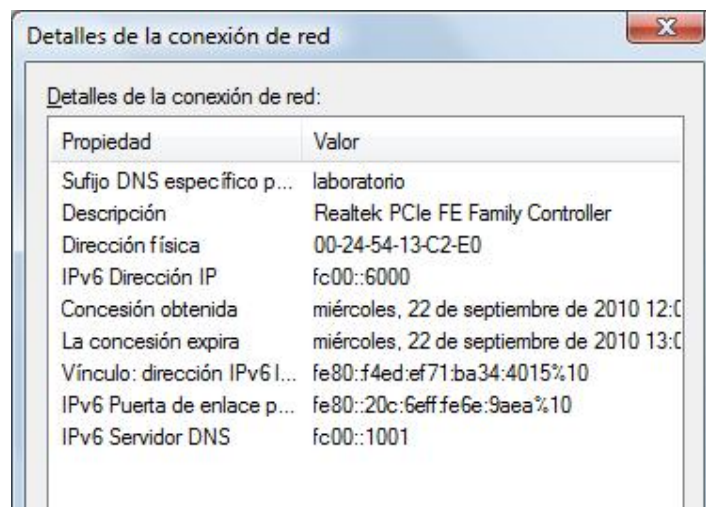
Adaptador de Ethernet Conexión de área local:

    Sufijo DNS específico para la conexión. . . : laboratorio
    Dirección IPv6 . . . . . : fc00::6000
    Vínculo: dirección IPv6 local. . . . . : fe80::f4ed:ef71:ba34:4015%10
    Puerta de enlace predeterminada . . . . . : fe80::20c:6eff:fe6e:9aea%10

C:\Users\josemiguel>_
```

Ilustración 7: Configuración de red Windows Vista (shell)

Se puede comprobar en la siguiente captura cómo el cliente ha tomado el servidor DNS que le envía el servidor:



Propiedad	Valor
Sufijo DNS específico p...	laboratorio
Descripción	Realtek PCIe FE Family Controller
Dirección física	00-24-54-13-C2-E0
IPv6 Dirección IP	fc00::6000
Concesión obtenida	miércoles, 22 de septiembre de 2010 12:00
La concesión expira	miércoles, 22 de septiembre de 2010 13:00
Vínculo: dirección IPv6 l...	fe80::f4ed:ef71:ba34:4015%10
IPv6 Puerta de enlace p...	fe80::20c:6eff:fe6e:9aea%10
IPv6 Servidor DNS	fc00::1001

Ilustración 8: Configuración de red Windows Vista (GUI)

5 Configuración y prueba de servicios

5.1 DNS

Como es sabido, el servicio de nombres de dominio resuelve consultas para averiguar la IP de un host a partir de un nombre. Si en IPv4 es manifiesta la utilidad del servicio DNS, todavía más cuando se trabaja con IPv6, con un formato de direcciones mucho más largo y de longitud variable (en el caso de que una persona la lea/escriba; un computador deberá tratarla en última instancia como un identificador de 128 bits).

Los registros que maneja un servidor DNS en el caso de IPv6 son del tipo AAAA, frente a los tipo A de IPv4.

La aplicación escogida para actuar como servidor para resolver peticiones DNS es BIND (Berkeley Internet Name Domain) BIND es, según el Internet Services Consortium, el software servidor de DNS más utilizado en Internet. Está presente en la distribución estable de Debian, y en nuestro caso lo instalamos en la máquina con dirección estática `fc00::1001`.

Con sistema operativo Debian:

```
# aptitude install bind9
```

Como ya se describió al inicio del documento, los 4 terminales se agrupan dentro de un dominio llamado `laboratorio.ln6`.

`pc1.laboratorio.ln6` → router

`pc2.laboratorio.ln6` → servidor DNS + servicios adicionales

cliente 1 → sistema Debian GNU/Linux

cliente 2 → sistema Windows Vista

De esta forma, se requiere configurar una zona en el servidor DNS llamada `laboratorio.ln6`. Para ello, debe editarse un fichero *ad hoc* dentro del directorio `/etc/bind`. Aunque existen proyectos para organizar registros de BIND en bases de datos tipo MySQL o PostgreSQL, en la configuración por defecto se guardan los parámetros y los registros DNS en ficheros de texto plano. Dada la envergadura de la red objeto de estudio, es suficiente con añadirlos a dichos ficheros de texto.

La dimensión (o la inmensidad) del conjunto de dominios en Internet exige dividirlos jerárquicamente para que cada organización informe adecuadamente al resto de las asociaciones host-IP de su propia red. En nuestro caso, teniendo una LAN que sirve peticiones a los clientes de la red, podemos establecer un TLD (Dominio de primer nivel)

ficticio llamado ln6. No es un dominio reconocido por la ICANN⁸, pero será funcional dentro de la red y, precisamente por no ser un TLD reconocido, podemos asegurarnos que las consultas las resuelve el servidor DNS de nuestra propia red.

Dentro de nuestra red, el servidor DNS resolverá las peticiones con carácter autoritativo para que no se delegue la tarea a otros servidores, como pudieran ser los de algún ISP o los propios Root Servers.

Como hemos dicho anteriormente, necesitamos crear una zona para nuestro dominio laboratorio.ln6. Para ello, creamos el fichero `/etc/bind/zone.ln6.laboratorio` con el siguiente contenido:

```
$TTL      604800
@         IN      SOA      laboratorio. ln6.laboratorio. (
                        2          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )    ; Negative Cache TTL

@         IN      NS       localhost.
@         IN      A        127.0.0.1
@         IN      AAAA     ::1

pc3      AAAA     fc00::4000
pc2      AAAA     fc00::1001
pc1      AAAA     fc00::1000
```

También es necesario indicar al servidor BIND que incluya esta nueva zona. Se debe editar el fichero `/etc/bind/named.conf.local` para que contenga:

```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used
in your
// organization
//include "/etc/bind/zones.rfc1918";
```

⁸<http://data.iana.org/TLD/tlds-alpha-by-domain.txt>

```
zone "laboratorio.ln6" {
    type master;
    file "/etc/bind/zone.ln6.laboratorio";
};
```

Una vez añadido el dominio de nuestra red local, basta reiniciar el servidor para que aplique los cambios y pueda resolver las peticiones que envían los clientes. Lo reiniciamos con la siguiente orden:

```
# /etc/init.d/bind9 restart
```

Ya desde el cliente 1, podemos comprobar que, efectivamente, somos capaces de averiguar una IPv6 a partir del nombre DNS, mediante la orden `host`:

```
josemiguel@localrdcx08:~$ host pc2.laboratorio.ln6
pc2.laboratorio.ln6 has IPv6 address fc00::1001
josemiguel@localrdcx08:~$ host pc1.laboratorio.ln6
pc1.laboratorio.ln6 has IPv6 address fc00::1000
```

Otra prueba adicional se hará cuando, al configurar el servicio HTTP, se compruebe desde los clientes que es posible visitar una web alojada en `pc2.laboratorio.ln6` (se requiere en ese caso, la pertinente consulta DNS, que será transparente para el usuario)

5.2 FTP

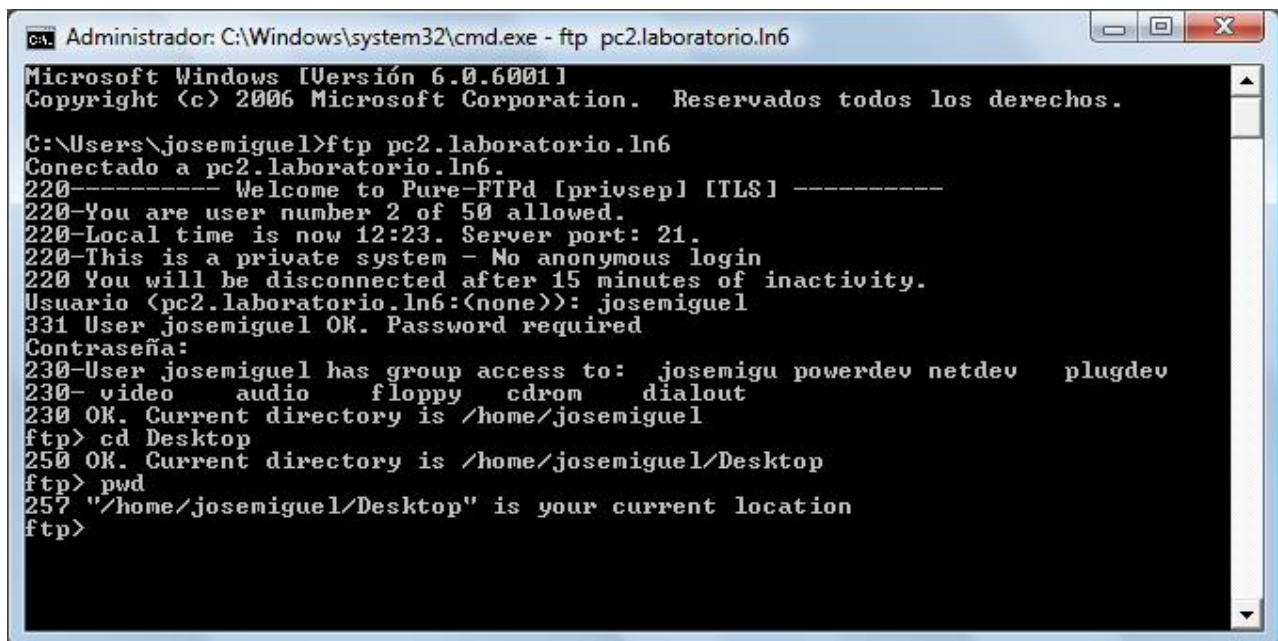
Para implantar el servicio FTP (File Transfer Protocol) en nuestra red, instalaremos el servidor Pure-FTPd. En Debian, desde la máquina con dirección IP `fc00::1001`, ejecutamos:

```
# aptitude install pure-ftpd
```

El servicio FTP requiere establecer una política de acceso (control de usuarios autorizados para operar con el servidor). El servidor pure-ftpd permite autorizar a usuarios de un dominio LDAP, así como a usuarios integrados en una BBDD o los propios del sistema UNIX donde esté instalado. Para esta última opción, basta con actualizar el fichero correspondiente mediante la orden:

```
# echo "yes" > /etc/pure-ftpd/conf/UnixAuthentication
```

Ya sea reiniciando el host o únicamente el servidor FTP, podemos probar la conexión desde el cliente Windows Vista:



```
Administrador: C:\Windows\system32\cmd.exe - ftp pc2.laboratorio.ln6
Microsoft Windows [Versión 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. Reservados todos los derechos.

C:\Users\josemiguel>ftp pc2.laboratorio.ln6
Conectado a pc2.laboratorio.ln6.
220----- Welcome to Pure-FTPd [privsep] [TLS] -----
220-You are user number 2 of 50 allowed.
220-Local time is now 12:23. Server port: 21.
220-This is a private system - No anonymous login
220 You will be disconnected after 15 minutes of inactivity.
Usuario (pc2.laboratorio.ln6:(none)): josemiguel
331 User josemiguel OK. Password required
Contraseña:
230-User josemiguel has group access to: josemiguel powerdev netdev plugdev
230- video audio floppy cdrom dialout
230 OK. Current directory is /home/josemiguel
ftp> cd Desktop
250 OK. Current directory is /home/josemiguel/Desktop
ftp> pwd
257 "/home/josemiguel/Desktop" is your current location
ftp>
```

Ilustración 9: Conexión por FTP desde Windows Vista (shell)

Podemos comprobar que el servidor escucha en el puerto 21, examinando las conexiones activas:

```
localrdcx08:/etc# netstat -6n
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp6      0      0 fc00::1001:21         fc00::6000:49172       ESTABLISHED
```

```

tcp6      0      0  ::1:33657          ::1:21             ESTABLISHED
tcp6      0      0  ::1:21             ::1:33657         ESTABLISHED
tcp6      0      0  fc00::1001:80     fc00::6000:49171  FIN_WAIT2

```

Comprobamos que también se puede acceder por interfaz gráfica (cliente gFTP) desde el mismo servidor (conectando a la dirección ip6-localhost, que es ::1)

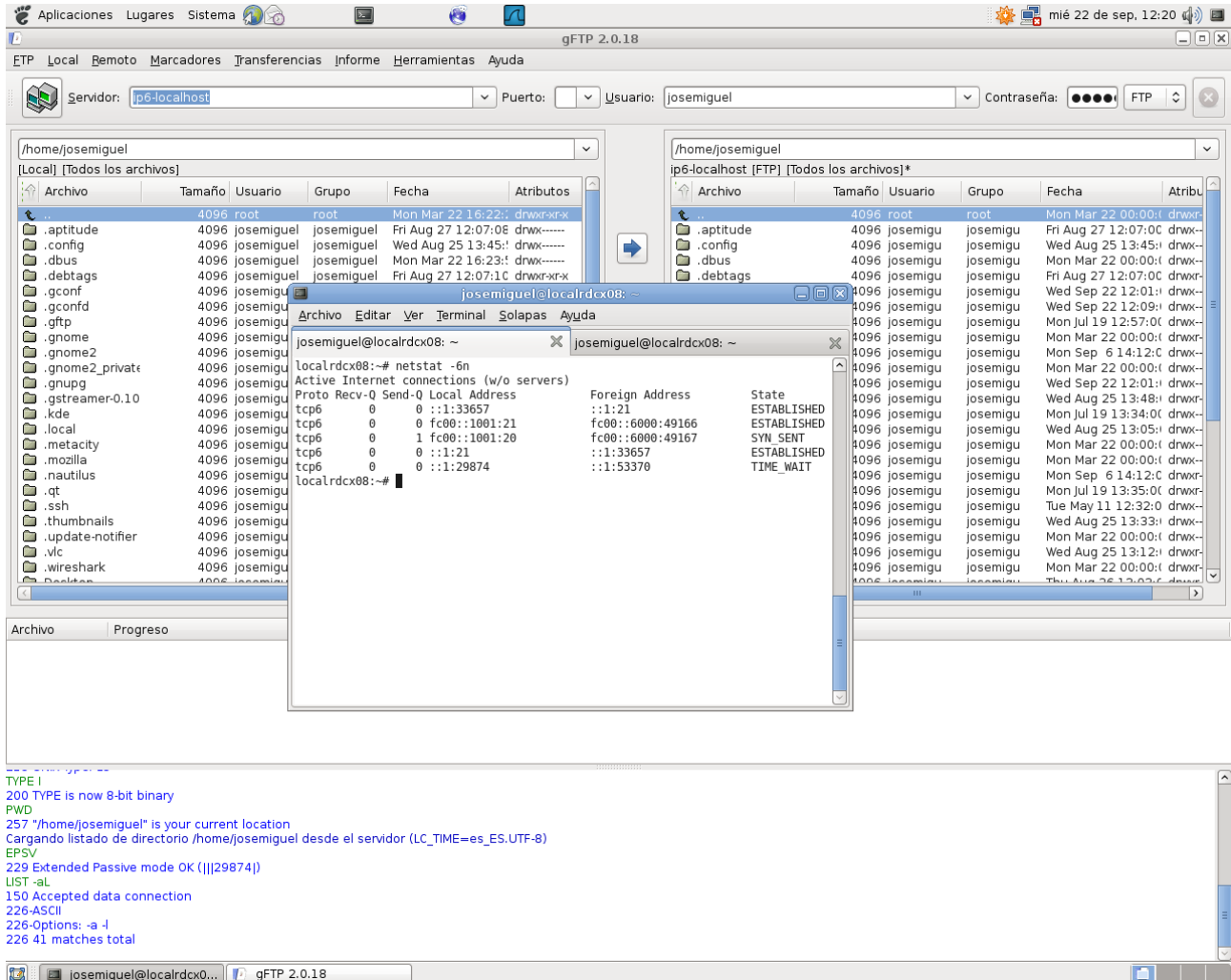


Ilustración 10: Sesión de gFTP

5.3 Web (HTTP)

Para ofrecer el servicio HTTP en la red local, utilizaremos el conocido servidor Apache, que podemos instalar con la siguiente orden:

```
# aptitude install apache2
```

Una vez instalado y lanzado el proceso servidor, dentro del DocumentRoot por defecto (directorio `/var/www/`) podemos editar `index.html` para elaborar una web de prueba. Tras modificar el fichero, hacemos la prueba desde los navegadores de los clientes. Se introduce la dirección `pc2.laboratorio.ln6` en la barra del navegador y, con ello, se comprueba el correcto funcionamiento de los servicios DNS y HTTP. Se hace la prueba desde Internet Explorer bajo Windows Vista y el resultado es el siguiente:

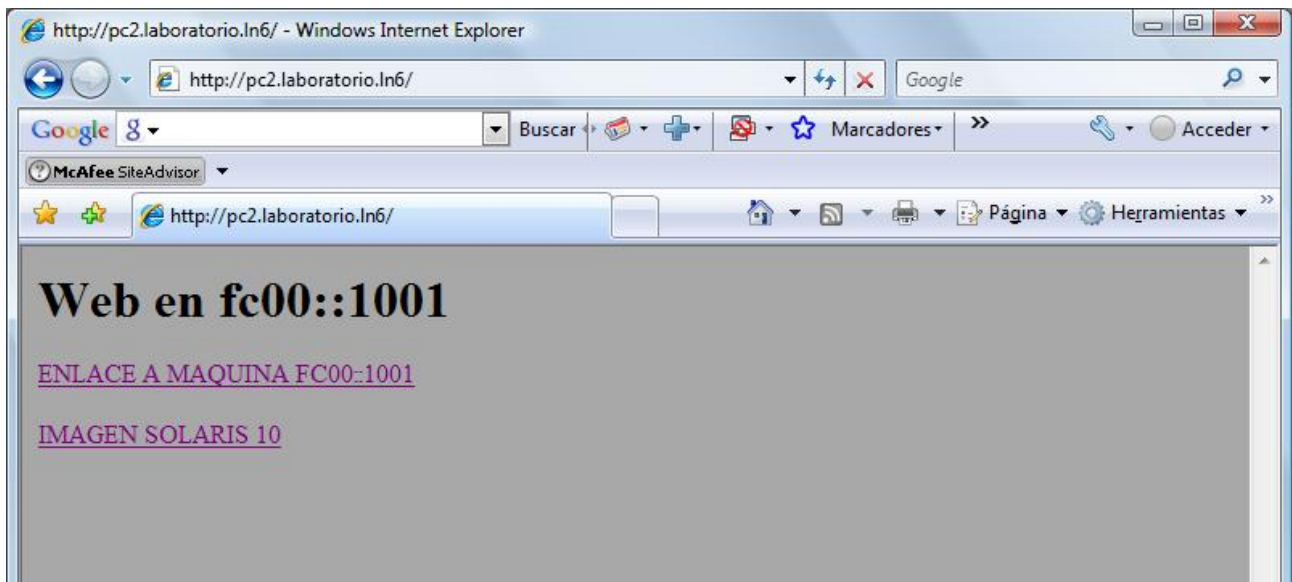


Ilustración 11: Consulta de web desde Internet Explorer

La web contiene dos enlaces, el primero de ellos a la dirección `http://[fc00::1001]/index2.html`. Visitando el primer enlace desde el cliente con sistema Debian, comprobamos que funciona:

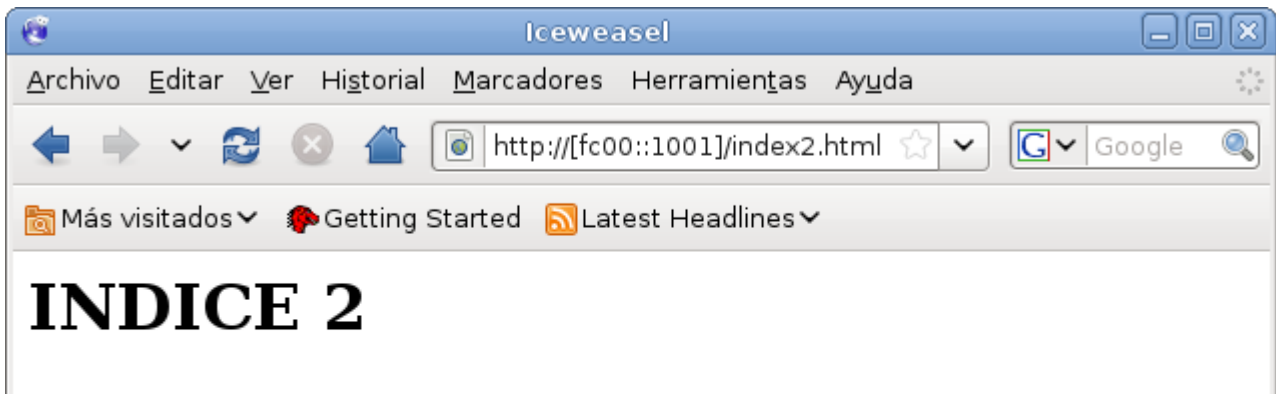


Ilustración 12: Consulta de web especificando IP

Como se puede observar en la barra de direcciones, es necesario introducir en el navegador la dirección IP entre corchetes. De otro modo, emite un aviso de que es una dirección para fines distintos a la navegación. Desde el mismo cliente Debian, comprobamos el código HTML de index.html, vía telnet al puerto 80:

```
josemiguel@localrdcx08:~$ telnet -6 fc00::1001 80
Trying fc00::1001...
Connected to fc00::1001.
Escape character is '^]'.
GET /
<html><body bgcolor="#a8a8a8"><h1>Web en
fc00::1001</h1></body>

<p><a href="http://[fc00::1001]/index2.html">ENLACE A MAQUINA
FC00::1001 </a></p>

<p> <a href="sol-10-u9-ga-x86-dvd.iso"> IMAGEN SOLARIS 10
</a> </p>
</html>
Connection closed by foreign host.
```


5.4 Vídeo en multicast

Se ha probado también en la red local objeto de estudio, la difusión de vídeo en streaming desde un nodo hacia el resto de la red.

El objetivo es emitir vídeo desde un nodo hacia un conjunto de direcciones, y que el cliente que lo desea se suscriba a la emisión, ejecutando un reproductor multimedia que obtenga el flujo de audio/vídeo vía red, de forma remota.

El contenido audiovisual se sirve desde el origen en streaming. La mayoría de servicios de vídeo en la red (tipo Youtube, Dailymotion, etc.) ofrecen el contenido bajo demanda. Es decir, el usuario abre una web con un objeto (normalmente Flash) e inicia la reproducción del mismo bajo demanda. Si la conexión es lo bastante rápida, el usuario puede acceder al contenido en cualquier momento. Por contra, emitiendo el vídeo desde el servidor mediante streaming, el reproductor del cliente capta el vídeo que en ese momento se esté transmitiendo desde el origen, de forma similar a una emisión tipo radio o TV. De hecho, en este modo de transmisión, no se espera que el cliente maneje una barra de tiempo para controlar el instante que desea reproducir. Más bien, recibe un flujo de audio y vídeo al ritmo que emita el servidor, de forma que éste marca el inicio, final y cambios de instante en la reproducción.

Para iniciar la reproducción en streaming, se requiere:

- Contenido multimedia para emitir
- Aplicación en el servidor para reproducir
- Aplicación en el cliente para decodificar el contenido
- Infraestructura de red (en nuestro caso, una LAN pero también a través de Internet)
- Dirección IP multicast
- Número de saltos (el número de routers que propagarán la emisión en multicast)

El contenido puede ser cualquier vídeo que acepten ambos reproductores (cliente y servidor). En este caso, optamos por un documental emitido en La2 sobre software libre, alojado en el FTP de la asociación PoLinux.

El nodo de la red que emitirá el vídeo será pc2, con IP `fc00::1001`, y en las pruebas se recibirá el stream desde el cliente 1, cuya IP se obtiene dinámicamente en el rango `fc00::6000 .. fc00::8000`.

El software empleado en ambos extremos, por su versatilidad reproduciendo muchísimos formatos y codecs y por sus capacidades para emitir/recibir contenidos vía red, es VideoLAN. Es software libre con licencia GPL y se puede obtener en www.videolan.org/vlc.

En cuanto a la dirección IP multicast, se opta por la dirección especial `ff08::1`. En la

práctica, en el servidor se abrirá un fichero con el vídeo a emitir, y se especificará que es un volcado en red hacia la dirección multicast. Además, se debe especificar el número de hops o routers que deberán propagar la emisión, para determinar el alcance de la misma.

La orden necesaria para empezar a emitir el vídeo es:

```
~vlc -v uned.avi --ipv6 --sout udp:[ff08::1] --ttl 2
```

Observamos que empieza la emisión, sin que se muestre por pantalla el propio vídeo (hemos elegido salida hacia la red y no por pantalla):

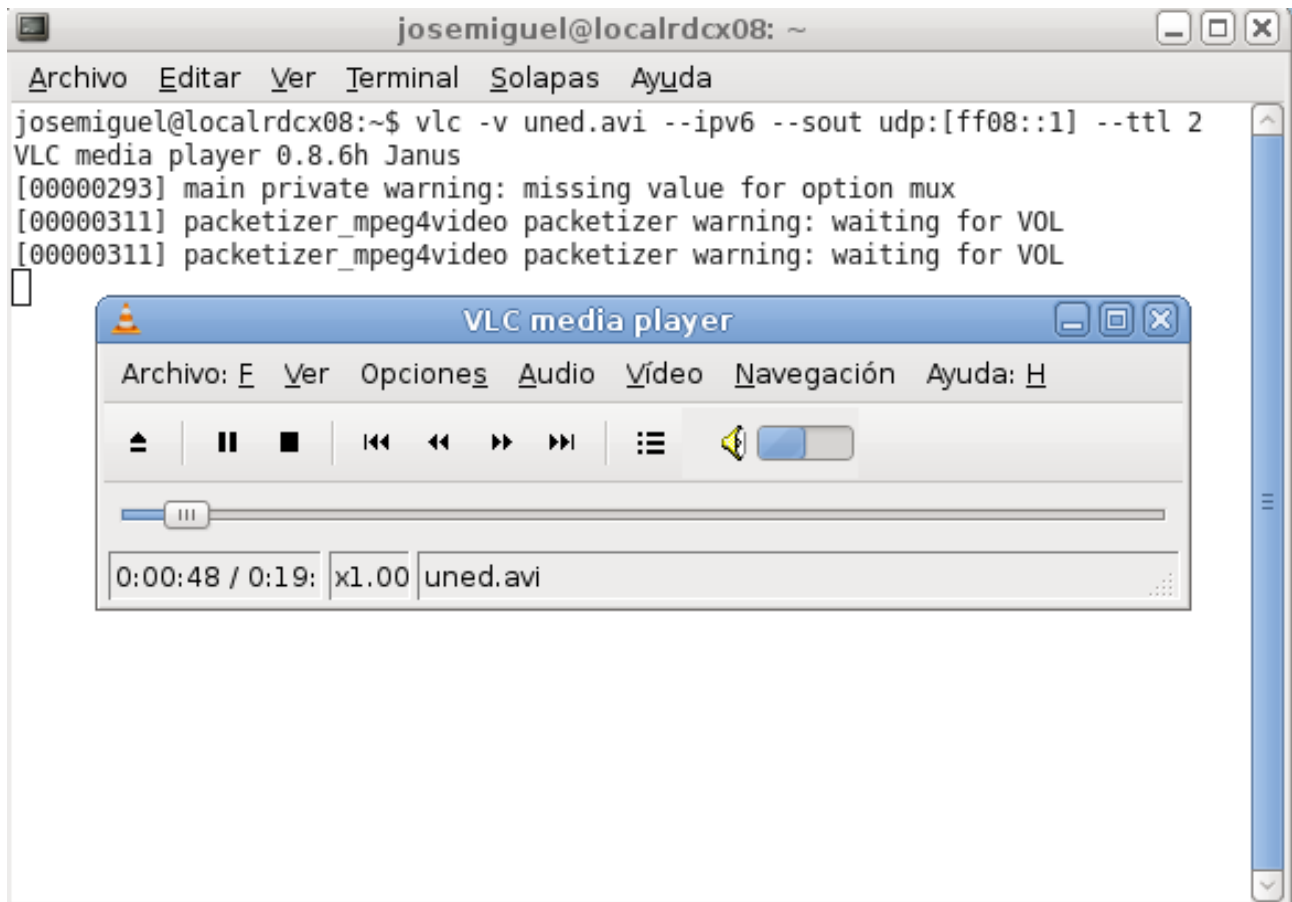


Ilustración 13: Streaming desde el servidor con VLC

Desde el cliente, la orden a ejecutar es:

```
~vlc --ipv6 udp:@\[ff08::1\]
```

Tras un corto tiempo de espera, el cliente empieza a recibir el flujo y lo puede visualizar y escuchar. Al ser una emisión en streaming, el cliente no tiene opción de controlar la reproducción más que para detenerla; no dispone de una barra de tiempo para acceso directo, sino que es el servidor quien controla qué contenido envía a todos los clientes que accedan a la dirección `ff08::1`. Se puede apreciar en la siguiente captura. La captura se tomó con cámara fotográfica, pues desde el sistema operativo devolvía un marco en negro.

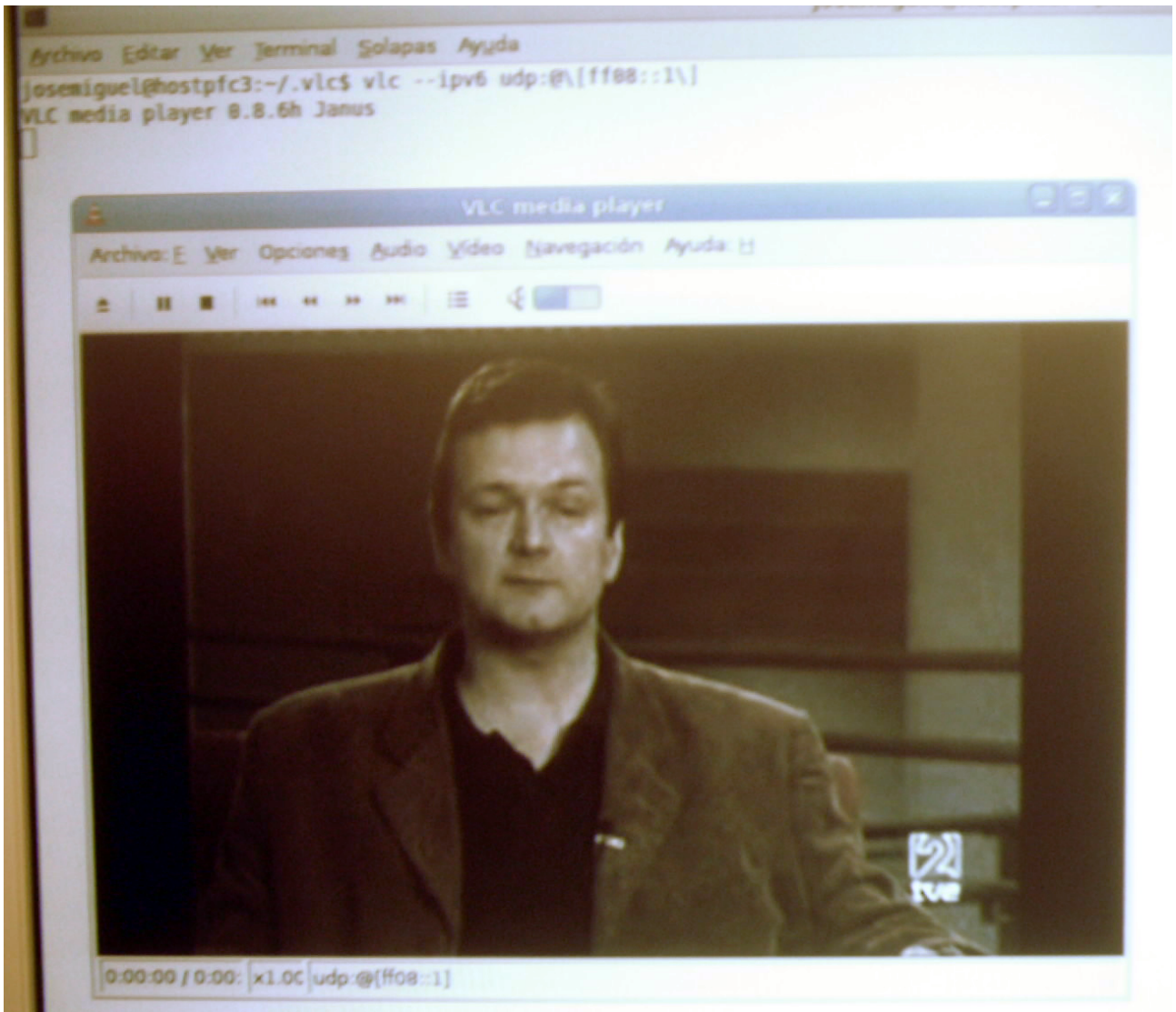


Ilustración 14: Vídeo en streaming desde cliente con VLC

5.5 SSH/SCP

Otro de los servicios ofrecidos en la red es el de Secure Shell (SSH), para administración remota de hosts en un canal seguro. SSH también permite la transferencia de ficheros bajo el canal seguro. Para instalar el servicio en el sistema Debian:

```
# aptitude install openssh-server
```

Haciendo una exploración de puertos del host con dirección `fc00::1001`, se observa que el servidor SSH escucha en el puerto 22.

```
josemiguel@localrdcx08:~$ nmap -6 fc00::1001
```

```
Starting Nmap 4.62 ( http://nmap.org ) at 2010-09-22 13:03 CEST
```

```
Interesting ports on fc00::1001:
```

```
Not shown: 1710 closed ports
```

```
PORT      STATE SERVICE
```

```
21/tcp    open  ftp
```

```
22/tcp    open  ssh
```

```
53/tcp    open  domain
```

```
80/tcp    open  http
```

```
143/tcp   open  imap
```

```
Nmap done: 1 IP address (1 host up) scanned in 0.155 seconds
```

Una vez que comprobamos que el servicio está lanzado y a la escucha, procedemos a realizar la prueba de conexión desde el cliente 1:

```
hostpfc3:~# ssh -6 josemiguel@fc00::1001
```

```
josemiguel@fc00::1001's password:
```

```
Linux localrdcx08.redes.upv.es 2.6.26-2-686 #1 SMP Mon Jun 21 05:58:44 UTC 2010 i686
```

```
The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
```

```
You have new mail.
```

```
Last login: Wed Sep 22 13:02:09 2010 from fc00::6001
```

Se puede capturar el tráfico generado entre cliente (fc00::6001) y servidor (fc00::1001) mediante Wireshark:

Time	Source	Destination	Protocol	Info
1 0.000000	fc00::6001	fc00::1001	SSH	Encrypted request packet len=48
2 0.000229	fc00::1001	fc00::6001	SSH	Encrypted response packet len=48
3 0.000342	fc00::6001	fc00::1001	TCP	36871 > ssh [ACK] Seq=49 Ack=49 W:
4 0.107447	fc00::1001	fc00::6001	SSH	Encrypted response packet len=80
5 0.107585	fc00::6001	fc00::1001	TCP	36871 > ssh [ACK] Seq=49 Ack=129

Ilustración 15: Captura de tráfico durante sesión SSH

Y, aprovechando la sesión abierta en el servidor SSH, comprobar que efectivamente la conexión queda establecida con el cliente:

```
josemiguel@localrdcx08:~$ netstat --inet -6n
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp6      1      0  ::1:33657              ::1:21                  CLOSE_WAIT
tcp6      0      0 fc00::1001:22         fc00::6001:36871      ESTABLISHED
```

También se comprueba desde el cliente 1 (desde otra sesión distinta) que la transferencia de ficheros vía SCP está operativa. En este caso, vamos a transferir un par de imágenes de sistemas operativos (es necesario incluir la IP entre corchetes):

```
hostpfc3:~# scp -6 josemiguel@[fc00::1001]:/var/www/*iso .
The authenticity of host 'fc00::1001 (fc00::1001)' can't be
established.
RSA key fingerprint is
31:e2:51:42:49:a7:1e:fc:3a:c9:15:59:a8:33:ba:ed.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'fc00::1001' (RSA) to the list of
known hosts.
josemiguel@fc00::1001's password:
PCBSD8.1-x86-DVD.iso
100% 1427MB 11.0MB/s 02:10
sol-10-u9-ga-x86-dvd.iso
```

100% 2048MB 11.0MB/s 03:06

6 Conclusiones

Se ha podido comprobar que es factible implantar los servicios básicos y bien conocidos en sus versiones para IPv6, con la salvedad de que hacía falta configurar el modem/router accediendo a una dirección IPv4. Una vez configurado, es cierto que IPv4 continúa apareciendo de forma testimonial en algún caso muy concreto (veíamos el script de Debian que intentaba localizar algún servidor DHCPv4), aunque es posible asignar direcciones IPv6 tanto a clientes como a servidores, y el comportamiento de las aplicaciones sigue siendo el esperado.

De todas formas, aunque las aplicaciones soporten perfectamente el protocolo IPv6, es necesario forzar expresamente el uso del protocolo mediante algún parámetro del tipo “-6” por línea de órdenes, o utilizando corchetes en la GUI del navegador. Esto puede confundir a muchos usuarios si deben suponer que el problema para comunicar con un servidor es por usar la versión 4 (cuya existencia probablemente desconozcan) en lugar de la versión 6. Es por esto que las transiciones desde IPv4 a IPv6 se están haciendo de forma escalonada y por el momento no se habla demasiado de la adopción de la nueva versión; si acaso, de que va a ser necesaria en algún momento.

En cuanto al soporte por parte de los sistemas operativos (no ya de las aplicaciones, una vez conseguida la integración en la red), se puede decir que, en general, es bueno. Como cliente, el más sencillo de configurar ha sido Windows Vista; solamente requirió deshabilitar IPv4 y establecer dirección y DNS automáticos para IPv6, ajustándose además al comportamiento marcado por el servidor (stateful o stateless). En los sistemas Debian, está perfectamente soportado en la parte de aplicaciones de tipo servidor; no obstante, en el cliente con idéntico SO debe editarse la interfaz de red para que busque servidores DHCPv6 utilizando un cliente (wide-dhcpv6-client) que por defecto no está instalado. Mención aparte para el sistema FreeBSD, cuya configuración IPv4 por DHCP fue sencilla e inmediata; en cambio, se consiguió obtener dirección IPv6 a partir de un prefijo ofrecido por radvd, pero no fue posible que se configurase mediante DHCPv6 stateful (aun observando el tráfico de red y viendo que se enviaban las correspondientes direcciones de host y DNS). En ese sentido, sería conveniente que los sistemas operativos contemplasen como probable el caso de un cliente que desee empezar a operar en una red IPv6 exclusiva.

Tampoco cabe esperar una migración completa desde IPv4 a IPv6. Existen en las empresas aplicaciones y sistemas operativos en producción configurados exclusivamente para IPv4, y que probablemente nunca necesiten enviar tráfico más allá de su red interna. Cambiar el tipo de protocolo para este software puede llegar a tener un precio muy alto, o directamente prohibitivo. También es frecuente el caso en que ni siquiera se dispone del código fuente para realizar modificaciones. En estos supuestos continuará siendo necesario el conocimiento de IPv4 para mantener la interoperabilidad de estos sistemas. Y llegado el momento en que necesiten salir al exterior, pasando por redes IPv6, deberá configurarse el túnel correspondiente (utilizando mecanismos como túneles 6in4, que encapsulan IPv4 dentro de IPv6).

De IPv6 podemos esperar mayor facilidad para obtener direcciones IP públicas (el conjunto total es de una dimensión gigantesca; $3,4 \cdot 10^{38}$ posibles) y un mejor soporte según aplicaciones (streaming multimedia frente a descarga de texto e imágenes estáticas). También está concebido para facilitar a los routers sus tareas de procesamiento de cabeceras y encaminamiento. Por otra parte, dadas sus nuevas características de seguridad, debería terminar con algunos problemas como el envío de tráfico en claro o técnicas como el spoofing.

Para empezar a probar IPv6 existen recursos en la web para iniciarse con este protocolo. Algunas webs aceptan conexiones utilizando ambos protocolos (versiones 4 y 6) y, para aquellos usuarios cuyo ISP sólo les facilite acceso vía IPv4, existen servicios del tipo tunnel brokers (SIXXS o Hurricane Electric) para integrarse en infraestructuras IPv6, con utilidades y manuales de ayuda.

7 Bibliografía

- TCP-IP. Arquitectura, protocolos e implementación con IPv6 y seguridad de IP
Feit, Sidnie
Madrid [etc.] : McGraw-Hill/Interamericana de España , 1998
- Oracle IPv6 Administration Guide
<http://dlc.sun.com/pdf/817-0573/817-0573.pdf>
- IPv6 Interface Identifiers
<http://msdn.microsoft.com/en-us/library/aa915616.aspx>
- 6Net Newsletter nº8
<http://www.6net.org/publications/newsletters/june-2005.pdf>
- radvd.conf Linux Man Page
<http://linux.die.net/man/5/radvd.conf>
- The TCP/IP Guide: IPv6/IPv4 Address Embedding
http://www.tcpipguide.com/free/t_IPv6IPv4AddressEmbedding-2.htm
- Internet Protocol, Version 6 (IPv6) Specification
<http://www.faqs.org/rfcs/rfc2460.html>
- The IPv6 Header and How it Works
<http://ipv6.com/articles/general/IPv6-Header.htm>
- IPv6 Addressing Architecture (RFC 3513)
<http://www.ietf.org/rfc/rfc3513.txt>
- IPv6 Multicast Address Space Registry
<http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xml>
- IPv6 Interface Identifiers
<http://msdn.microsoft.com/en-us/library/aa915616.aspx>
- Microsoft IPv6
<http://technet.microsoft.com/en-us/network/bb530961.aspx>
- IPv6 Stateless Address Configuration
<http://www.ietf.org/rfc/rfc2462.txt>
- Privacy Extensions for Stateless Address Autoconfiguration

<http://www.faqs.org/rfcs/rfc3041.html>

- Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
<http://www.faqs.org/rfcs/rfc3315.html>
- Default Address Selection for Internet Protocol Version 6 (IPv6)
<http://www.ietf.org/rfc/rfc3484.txt>
- BIND for the small LAN
<http://www.madboa.com/geek/soho-bind/>
- Autoconf and DHCPv6
http://lacnic.net/documentos/lacnicxi/presentaciones/autoconf_and_dhcpv6.pdf
- Streaming_HowTo/Streaming_over_IPv6
http://wiki.videolan.org/Documentation:Streaming_HowTo/Streaming_over_IPv6