

Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros de Telecomunicación



**DISEÑO DE ESCENARIOS DE TRANSICIÓN A IPV6
UTILIZANDO LA HERRAMIENTA VNX:
ENCAMINAMIENTO Y ASIGNACIÓN DE
DIRECCIONES**

TRABAJO FIN DE MÁSTER

Rafael García García

2011

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación

**Máster Universitario en
Ingeniería de Redes y Servicios Telemáticos**

TRABAJO FIN DE MÁSTER

**DISEÑO DE ESCENARIOS DE TRANSICIÓN A IPV6
UTILIZANDO LA HERRAMIENTA VNX:
ENCAMINAMIENTO Y ASIGNACIÓN DE
DIRECCIONES**

Autor

Rafael García García

Director

David Fernández Cambroner

Departamento de Ingeniería de Sistemas Telemáticos

2011

Resumen

Cuando utilizamos Internet para cualquier actividad, ya sea correo electrónico, navegación web, descarga de ficheros, o cualquier otro servicio o aplicación, la comunicación entre los diferentes elementos de la red y nuestro propio ordenador o teléfono, utiliza un protocolo que denominamos Protocolo de Internet (IP, Internet Protocol).

En los últimos años, prácticamente desde que Internet tiene un uso comercial, la versión de este protocolo es la número 4 (IPv4).

Para que los dispositivos se conecten a la red, necesitan una dirección IP. Cuando se diseñó IPv4, casi como un experimento, no se pensó que pudiera tener tanto éxito comercial, y dado que sólo dispone de 2^{32} direcciones (direcciones con una longitud de 32 bits, es decir, 4.294.967.296 direcciones), junto con el imparable crecimiento de usuarios y dispositivos, implica que este número resulta claramente insuficiente.

A raíz de la entrega por parte del IANA de los últimos paquetes con direcciones de Internet IPv4 en Febrero de 2011, la necesidad de una transición al protocolo llamado a sustituirle, el protocolo IPv6, se hace cada vez más patente, especialmente en los ámbitos de docencia e investigación técnica que deben ser los pioneros en estos procesos.

En este Trabajo Fin de Máster vamos a desarrollar un escenario simulado que nos permita modelar la transición del protocolo de red IPv4 al protocolo IPv6 en una escuela de la Universidad Politécnica de Madrid (UPM en adelante), en concreto en la Escuela de Ingeniería Aeronáutica y del Espacio (EIAE en adelante), una escuela que conocemos y que demanda cada vez más la transición al nuevo protocolo de red para promover la investigación y mejorar las actividades docentes.

Utilizaremos, para realizar esta simulación, una herramienta desarrollada por el Departamento de Ingeniería de Sistemas Telemáticos de la Escuela Técnica Superior de Ingenieros de Telecomunicación llamada VNX/VNUML que nos permitirá generar el escenario completo con el único requisito de disponer de una máquina anfitriona relativamente potente en cuanto a memoria, capacidad de proceso y almacenamiento.

Finalmente cabe resaltar que esta memoria forma parte de un trabajo conjunto realizado con Gloria Martín Martín en el que se ha desarrollado el escenario completo,

por lo que ambos trabajos presentan una parte común (introducción y descripciones generales) y una parte específica que se describe en cada una de las dos memorias. Este Trabajo Fin de Máster realiza la descripción de una parte del escenario completado en el Trabajo Fin de Máster de Gloria Martín Martín donde se describe el resto de servicios y configuraciones que lo componen.

Abstract

When we use the Internet for any activity, whether email, web browsing, downloading files, or any other service or application, communication between different network elements and our own computer or telephone, is using a protocol called Internet Protocol (IP, Internet Protocol).

In recent years, practically since the Internet has a commercial use, the version of this protocol is the number 4 (IPv4).

For devices to connect to the network they need an IP address. When IPv4 was designed, almost as an experiment, it was not thought that it could have so much commercial success, and since it only have 2^{32} addresses (addresses with a length of 32 bits, 4.294.967.296 addresses), along with the unstoppable growth of users and devices, means that this number is clearly insufficient.

Following the delivery by the IANA of the latest packages with IPv4 Internet addresses in February 2011, the need for a transition to the protocol designed to replace IPv4, the IPv6 protocol, it becomes increasingly evident, especially in the areas of teaching and technical research which are called to be the pioneers in these processes.

In this Master's Thesis we will develop a simulated scenario that will allow us to model the transition from IPv4 network protocol to IPv6 in a school of the Polytechnic University of Madrid (UPM), particularly in the College of Engineering Aeronautics and Space (EIAE) a school we know and demand increasingly a transition to the new network protocol to promote research and improve teaching activities.

We will use to perform this simulation, a tool developed by the Department of Telematic Systems Engineering of the School of Telecommunications Engineers named VNX/VNUML. With this tool we will be able to generate the whole scenario with the only requirement to have a relatively powerful host machine for memory, processing power and storage.

Finally it should be noted that this memory is part of a joint work with Gloria Martín Martín in which we have developed the complete scenario, so that both jobs have a common part (introduction and general description) and a specific part that is described in each of the two memories. This Master's Thesis makes the description of a

part of the scenario completed in the Master's Thesis of Gloria Martín Martín, which describes the other services and settings that compose it.

Índice general

Resumen	i
Abstract.....	v
Índice general.....	vii
Índice de figuras	ix
Siglas	xi
1 Introducción.....	1
2 IPv4 vs IPv6.....	3
2.1 Protocolos de red.....	3
2.2 IPv4.....	3
2.2.1 Formato del datagrama IPv4	4
2.2.2 Limitaciones de IPv4.....	5
2.3 IPv6.....	5
2.3.1 Características principales.....	6
2.3.2 Formato del paquete	7
2.3.3 Direccionamiento en IPv6	9
2.4 Estrategias de transición a IPv6 (RFC1933)	12
2.4.1 Nodos duales o doble pila IPv4- IPv6 (Dual Stack)	13
2.4.2 Túneles	15
2.4.3 Traductores de protocolos.....	17
3 Virtualización y VNX.....	18

3.1	Virtualización.....	18
3.2	VNX/VNMUL.....	20
3.3	Funcionamiento de VNX.....	23
4	Migración a IPv6 en una escuela de la UPM (EIAE)	24
4.1	Introducción.....	24
4.2	Modelo de direccionamiento	24
4.3	Estrategia de transición	29
5	Prototipos	30
5.1	Simulador	30
5.2	Descripción del escenario.....	32
5.3	Servicios.....	39
5.3.1	Configuración del protocolo	39
5.3.2	Encaminamiento.....	48
5.3.3	Configuración Stateful.....	50
5.3.4	Firewall	65
6	Conclusiones	68
7	Referencias.....	69

Índice de figuras

Figura 1. Estructura de un paquete IPv4	4
Figura 2. Formato cabecera IPv6.....	7
Figura 3. Cabeceras IPv6	9
Figura 4. Estructura de las direcciones IPv6	10
Figura 5. Ejemplo de dirección IPv6.....	11
Figura 6. Transición de IPv6 a IPv4	12
Figura 7. Pila dual IPv4 e IPv6	13
Figura 8. Funcionamiento de la Pila Dual	14
Figura 9. Encapsulación de datagramas	15
Figura 10. Túnel IPv6 encapsulado en IP4.....	16
Figura 11. Funcionamiento del Traductor NAT64	18
Figura 12. Sistema Real (izquierda) y sistema virtualizado (derecha)	19
Figura 13. Funcionamiento básico de VNX	21
Figura 14. Virtualización con VNX.....	22
Figura 15. Funcionamiento de VNX.....	23
Figura 16. Plan de direccionamiento IPv6 en la EIAE	26
Figura 17. Escenario de simulación	32
Figura 18. Instalación del protocolo IPv6 en Windows XP	40
Figura 19. Configuración del protocolo IPv4 en Windows XP	42
Figura 20. Comprobación de la configuración IPv4 en “Xp-2”	45
Figura 21. Comprobación de la configuración IPv6 en “Xp-2”	45

Figura 22. Comprobación de la conectividad IPv4 en "Xp-2"	46
Figura 23. Comprobación de la conectividad IPv6 en "Xp-2"	46
Figura 24. Comprobación del enrutamiento en IPv4 en "Xp-2"	49
Figura 25. Comprobación del enrutamiento en IPv6 en "Xp-2"	50
Figura 26. Selección de paquetes a instalar en el asistente de Dibbler"	53
Figura 27. Configuración del protocolo IPv4 por DHCP en Windows XP	55
Figura 28. Configuración de los protocolos IPv4 e IPv6 en un interfaz de red en Windows XP.....	61
Figura 29. Comprobación de la conectividad a nivel IPv4 en un interfaz de red en Windows XP.....	61
Figura 30. Funcionamiento del sistema cliente de Dibbler en "Xp-1"	62
Figura 31. Configuración de IPv6 del sistema "Xp-1"	63
Figura 32. Comprobación de la conectividad en IPv6 desde "Xp-1" hasta "Srv-nat64"	63

Siglas

DHCP: Dynamic Host Configuration Protocol.

DNS: Domain Name System.

DNS64: Domain Name System 6 to 4.

FTP: File Transfer Protocol.

IPv6: Significa Internet Protocol version 6.

IPv4: Significa Internet Protocol version 4.

MAC: Media Access Control address.

NAT64: Network Address Translation 6 to 4.

RA: Router Advertisement.

RADVD: Router Advertisement Daemon.

UML: User mode Linux.

VLAN: Virtual Local Area Network.

VNUML: Virtualization Network User Mode Linux.

VNX: Virtual Networks over Linux.

XML: eXtensible Markup Language.

WWW: World Wide Web.

1 Introducción

El impulso del protocolo IPv6 se debe sobre todo a un motivo: la necesidad de más direcciones. Hoy en día hay millones de nuevos dispositivos como teléfonos móviles, PDAs, dispositivos de consumo, etc., algunos de los cuales necesitan más de una dirección IP. Además, se ha incrementado el número de usuarios incorporándose a Internet en países como China e India.

Inicialmente se buscaron soluciones que permitieran compartir direcciones utilizando mecanismos como NAT, PPP, etc., pero esto, aunque ha permitido el crecimiento de Internet, ha provocado problemas debido a la pérdida de conectividad extremo a extremo, lo que conlleva que algunas aplicaciones dejen de funcionar.

Finalmente se vio que la necesidad de un nuevo protocolo era ineludible, pues aún en el caso de una utilización más óptima de las direcciones IP, el número de ellas que proporciona IPv4 resulta insuficiente y el uso de NAT hace más costoso y complejo el desarrollo de servicios y aplicaciones y por lo tanto puede frenar la innovación en Internet.

Hoy en día IPv6 ya no es un protocolo en fase de diseño y experimentación y el trabajo técnico relacionado con este protocolo prácticamente está finalizado. Hoy todas las ventajas que introduce este nuevo protocolo (espacio virtualmente ilimitado de direcciones, seguridad a nivel de red, movilidad, multicast, etc.) aparecen disponibles ante un mundo que en el que se prevé que IPv6 genere una nueva ola de innovación en las aplicaciones y en la oferta de servicios ya que termina con la necesidad de direcciones compartidas.

La mayoría de sistemas operativos están preparados para el nuevo protocolo y la migración al mismo no debería suponer problemas al usuario final por eso, a pesar de que desde el punto de vista profesional existe aún cierto recelo a la introducción de esta nueva tecnología, la convergencia tecnológica a medio plazo hacia el nuevo protocolo IPv6 es inevitable.

La implantación de IPv6 no es, ni va a ser, inmediata. Además de todo el esfuerzo invertido en los últimos años en su desarrollo, ahora es necesario desplegar esta nueva tecnología en la red. Esto requiere actualizar el software y, en algunos casos, el hardware de millones de equipos y también conllevará cambios en las aplicaciones, procedimientos y herramientas de gestión de redes.

Hasta ahora han sido muy pocos los servicios y contenidos de Internet accesibles mediante IPv6, por lo que el tráfico IPv6 visible en la red es aún mínimo.

Es importante además que las Universidades y Centros de Investigación lideren el proceso de transición a esta nueva tecnología.

El objetivo de este proyecto es realizar, mediante una potente herramienta de virtualización como es VNX, un escenario simulado en el que realizar la transición de IPv4 al IPv6 en una Escuela de la UPM, tomando como objetivo la implantación de algunos de los servicios más relevantes que se ofrecen en la Universidad como pueden ser el servicio DNS, el servicio WEB, el servicio DHCP, etc.

A lo largo de este Trabajo Fin de Máster desarrollaremos un modelo en el que a través de varias redes, servidores y clientes instalaremos y configuraremos los diferentes servicios tanto en IPv4 como en IPv6. Así mismo realizaremos las pruebas pertinentes para comprobar el correcto funcionamiento de la red y los servicios con ambos protocolos.

También se estudiará uno de los mecanismos de transición para comunicar IPv4 e IPv6 durante el tiempo que necesiten coexistir. Este mecanismo será el mecanismo de los traductores de protocolos y en concreto se experimentará en el escenario propuesto con el traductor NAT64.

En este Trabajo Fin de Máster forma parte de un trabajo conjunto realizado con Gloria Martín Martín. En esta memoria se detallará una parte de los trabajos realizados, y el resto se explican en el Trabajo Fin de Máster de Gloria Martín Martín. Ambos trabajos se presentan conjuntamente por formar parte del mismo objetivo.

2 IPv4 vs IPv6

En este apartado vamos a describir brevemente los protocolos de Internet versión 4 (IPv4) y versión 6 (IPv6) así como los factores que han provocado la necesaria implantación de IPv6 como futuro protocolo de Internet.

2.1 Protocolos de red

La comunicación en las redes de información es posible gracias a los protocolos de red. Un protocolo es un conjunto de reglas que controla o permite la comunicación entre dos o más dispositivos mediante la transmisión y recepción de paquetes de bits con una cierta estructura. En Internet, el protocolo más utilizado es el TCP/IP (Transmission Control Protocol/Internet Protocol).

2.2 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 máquinas 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 re-ensamblar los distintos fragmentos recibidos para formar el datagrama original.

IPv4 es la versión 4 del Protocolo IP. Esta fue la primera versión del protocolo que se implementó extensamente, y forma la base de Internet. Fue publicado en 1981 en la RFC 791.

Tiene la capacidad de ofrecer más de 4.200 millones de direcciones, pero no está capacitado para cubrir la demanda actual existente, no sólo por el número de direcciones que soporta, sino también por la forma en que agrupa los bits en su sistema de numeración network/host (que desperdicia direcciones y sufre excesiva sobrecarga en el routing).

2.2.1 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).

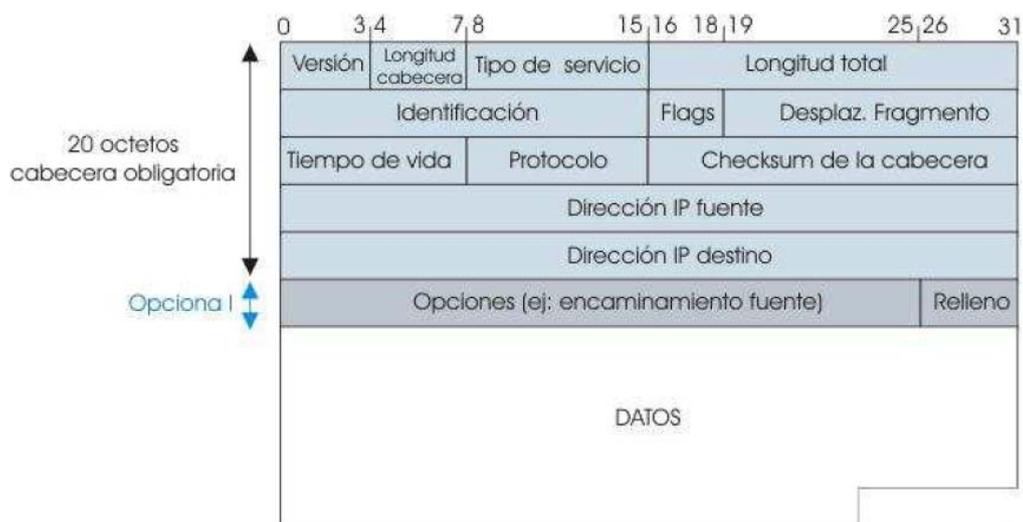


Figura 1. Estructura de un paquete IPv4

2.2.2 Limitaciones de IPv4

IPv4 usa direcciones de 32 bits, limitándose así a $2^{32} = 4.294.967.296$ direcciones. Esto, que en un principio puede parecer más que suficiente, en realidad no lo es tanto.

Además hay que tener en cuenta que no todas las combinaciones están disponibles para el protocolo IP público. Hay una serie de combinaciones reservadas lo que hace que el número real de IP's disponibles no sea tan elevado.

En principio hay que asegurar una serie de combinaciones diferentes para las conexiones a Internet, por lo que a cada proveedor ISP (Internet Service Provider) se le asigna un determinado número de direcciones IP, asignándolas estas a su vez entre sus clientes. Para optimizar este número de conexiones los proveedores ISP recurren al sistema de IP dinámica. Este sistema hace posible que con un número limitado de IP's se atienda a un número bastante superior de usuarios, a condición de que el número de conexiones simultáneas no supere el número de IP's asignadas.

Además hay que tener en cuenta que este tipo de conexiones es cada vez más empleado. No sólo por ordenadores, sino también por dispositivos de otro tipo, tales como, por ejemplo, cámaras IP, comunicaciones de voz del tipo VoIP, teléfonos móviles, PDA, etc., lo que ha provocado, junto al cambio de hábitos en las conexiones (hemos pasado de conexiones por un corto periodo de tiempo cuando nos conectábamos por RTB a tener conectado el ordenador las 24 horas, o al menos 8 horas diarias) y el incremento en el número de usuarios (que prácticamente se duplica cada año desde 1.988), que el número de conexiones disponibles no sólo no sea exagerado, sino que sea claramente insuficiente.

Una de las consecuencias de este sistema es la necesidad de utilizar para conectarse a la red (Internet) un sistema que permita una sola IP por conexión (independientemente de los ordenadores que luego se conecten a través de esta conexión). Este sistema es el denominado NAT (Network Address Translation), y permite mediante un router tener una red interna (direcciones IP privadas) apuntando a una sola dirección de Internet (IP pública).

Esta serie de limitaciones han ayudado a estimular el impulso hacia IPv6.

2.3 IPv6

IPv6 o IPng (Next Generation Internet Protocol) es la nueva versión del protocolo IP. Ha sido diseñado por el IETF (Internet Engineering Task Force) para reemplazar de forma gradual a la versión actual, el IPv4.

2.3.1 Características principales

IPv6 surge, sobre todo, para resolver los problemas de direccionamiento y encaminamiento detectados en los años 90 en Internet.

IPv6 es una evolución de IPv4, no es una revolución ya que no aporta grandes novedades, sino que aprovecha los años de experiencia que aporta IPv4.

Las características más importantes que aporta IPv6 son las siguientes:

- **Mayor espacio de direcciones.** El tamaño de las direcciones IP cambia de 32 bits a 128 bits, para soportar más niveles de jerarquías de direccionamiento y más nodos direccionables.
- **Simplificación del formato de la cabecera.** Algunos campos de la cabecera IPv4 se quitan o se hacen opcionales.
- **Posibilidad de paquetes con carga útil (datos)** de más de 65.355 bytes.
- **Seguridad en el núcleo del protocolo.** El soporte de IPsec es un requerimiento del protocolo IPv6.
- **Capacidad de etiquetas de flujo.** Puede ser usada por un nodo origen para etiquetar paquetes pertenecientes a un flujo de tráfico particular, que requieren manejo especial por los routers IPv6, tal como calidad de servicio o servicios de tiempo real.
- **Autoconfiguración** de direcciones es más simple. Especialmente en direcciones Agregatable Global Unicast, los 64 bits superiores son enviados por medio de un mensaje desde el router (Router Advertisement) y los 64 bits más bajos son obtenidos a partir de la dirección MAC (en formato EUI-64). En este caso, el largo del prefijo de la subred es 64, por lo que no hay que preocuparse más por la máscara de red. Además el largo del prefijo no depende en el número de los hosts por lo tanto la asignación es más simple.
- **Re-numeración y "multihoming",** facilitando el cambio de proveedor de servicios.
- **Características de movilidad,** posibilitando a un nodo mantener la misma dirección IP, a pesar de su movilidad. Frente a IPv4 se mejora la eficiencia y seguridad.
- **Enrutamiento más eficiente en el backbone de la red,** debido a la jerarquía de direccionamiento basada en agregación. El encaminamiento es jerárquico basado en la agregación de rutas.
- **Calidad de servicio (QoS) y clase de servicio (CoS).**
- **Capacidades de autenticación y privacidad.**

- **Multicast:** envío de un mismo paquete a un grupo de receptores.
- **Anycast:** envío de un paquete a un receptor dentro de un grupo.
- **Paquetes IP eficientes y extensibles,** sin que haya fragmentación en los routers, alineados a 64 bits (preparados para su procesamiento óptimo con los nuevos procesadores de 64 bits), y con una cabecera de longitud fija, más simple, que agiliza su procesamiento por parte del router.

2.3.2 Formato del paquete

El formato de la cabecera de un paquete IPv6 es el siguiente:



Figura 2. Formato cabecera IPv6

La longitud de esta cabecera es de 40 bytes, el doble que en el caso de IPv4, pero con muchas ventajas, al haberse eliminado campos redundantes.

Además, como ya hemos mencionado, la longitud fija de la cabecera, implica una mayor facilidad para su procesamiento en routers y conmutadores, incluso mediante hardware, lo que implica unas mayores prestaciones.

A este fin ayuda, como hemos indicado anteriormente, el hecho de que los campos están alineados a 64 bits, lo que permite que las nuevas generaciones de procesadores y microcontroladores, de 64 bits, puedan procesar mucho más eficazmente la cabecera IPv6.

- El **campo de versión**, que es igual a 6, lógicamente, tiene una longitud de 4 bits.
- **Clase de Tráfico**, también denominado Prioridad, o simplemente Clase. Tiene una longitud de 8 bits (1 byte).
- **Etiqueta de Flujo**, para permitir tráfico con requisitos de tiempo real. Tiene una longitud de 20 bits.

- **Longitud de la carga útil:** es la longitud de los propios datos, y puede ser de hasta 65.536 bytes. Tiene una longitud de 16 bits (2 bytes).
- **Siguiente cabecera:** dado que en lugar de usar cabeceras de longitud variable se emplean sucesivas cabeceras encadenadas, desaparece el campo de opciones. En muchos casos ni siquiera es procesado por los routers, sino tan sólo extremo a extremo. Tiene una longitud de 8 bits (1 byte). Este campo indica qué viene detrás de la cabecera y pueden ser paquetes TCP, cabeceras opcionales, etc.
- **Límite de saltos:** tiene una longitud de 8 bits (1 byte). No es necesario aumentarlo ya que las redes son jerárquicas y no se prevé que un paquete haga más de 256 saltos.

Las **cabeceras opcionales** se definen en el valor del campo “siguiente cabecera” e indica cual es la siguiente cabecera y así sucesivamente. Las sucesivas cabeceras, no son examinadas en cada nodo de la ruta, sino sólo en el nodo o nodos destinos finales. Hay una única excepción a esta regla: cuando el valor de este campo es cero, lo que indica opción de examinado y proceso “salto a salto” (hop-by-hop). Así tenemos, por citar algunos ejemplos, cabeceras con información de encaminado, fragmentación, opciones de destino, autenticación, cifrado, etc., que en cualquier caso, han de ser procesadas en el orden riguroso en que aparecen en el paquete.

Los tipos de cabeceras opcionales definidas son las siguientes:

- **Hop by Hop Options:** esta información debe ser examinada en cada salto.
- **Routing:** similar a la opción Source Route de IPv4.
- **Fragmentation:** Segmentación y re-ensamblado.
- **Authentication:** Firmas digitales.
- **Security Encapsulation:** Cifrado.
- **Destination Options:** Información examinada sólo en destino.

Sin entrar en más detalles, véanse a continuación los siguientes ejemplos gráficos del uso del concepto de las “cabeceras de extensión” (definidas por el campo “siguiente cabecera”), mecanismo por el que cada cabecera es “encadenada” a la siguiente y anterior (si existen):

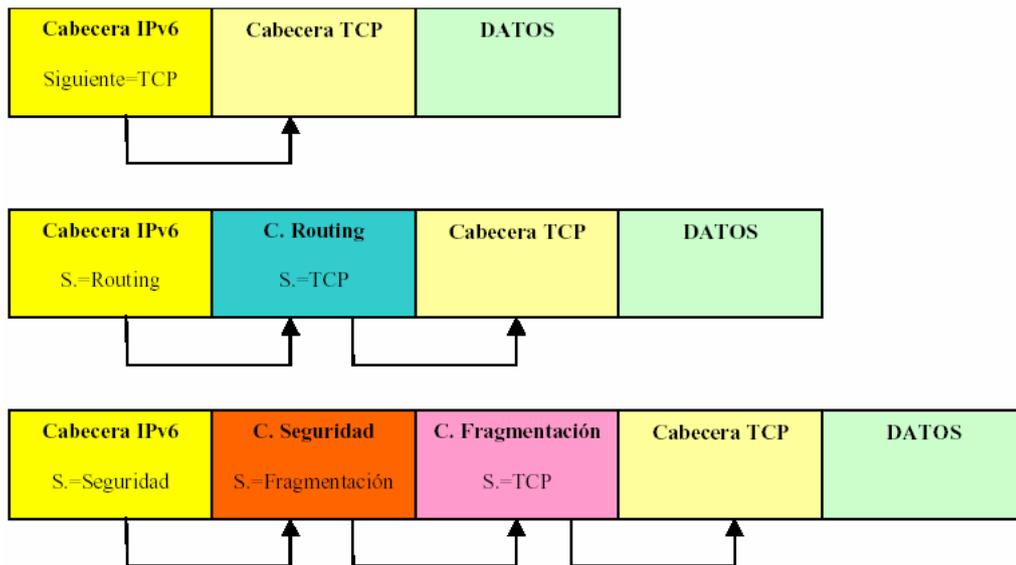


Figura 3. Cabeceras IPv6

La mejora del **formato del paquete IPv6** ofrece las siguientes ventajas:

- No hay limitaciones en el número de opciones.
- Mejora las prestaciones debido a la ordenación de la cabecera:
 - Cabeceras procesadas por los routers más hacia la derecha, hacia el exterior de la cabecera.
 - Cabeceras procesadas en destino, más hacia la izquierda, hacia el interior de la cabecera.

Las cabeceras opcionales están pensadas para facilitar el trabajo sobre todo a los routers ya que pueden saber qué tipo de opciones incluye la cabecera y ver si le interesa o no. Por ejemplo, a un router no le interesa una opción que sea una firma digital. La idea es que sólo analice lo que tiene que analizar porque le sea interesante para el proceso de encaminamiento de los paquetes.

2.3.3 Direccionamiento en IPv6

Modelo de direccionamiento

Las direcciones son de 128 bits e identifican interfaces individuales o conjuntos de interfaces. Al igual que en IPv4 los nodos se asignan a interfaces. De esta forma, podemos encontrar interfaces con una dirección asignada o interfaces con varias direcciones asignadas.

Cada máquina conectada a la red dispone de dos direcciones:

- **Dirección de ámbito local:** le permite comunicarse con los equipos que están en su misma red. Esta dirección se obtiene mediante la autoconfiguración.

- **Dirección de ámbito global:** le permite comunicarse con cualquier equipo conectado a la red Internet.

Las direcciones globales permiten a los sistemas conectarse a Internet, para ello se necesita obtener el prefijo global. Este prefijo lo obtienen los sistemas finales del router de la red ya que de forma periódica éste envía información del prefijo global. De esta forma, los equipos pueden conocer la dirección del router de la red y de la dirección global del sistema.

Las direcciones se clasifican en tres tipos:

- **Unicast**, que identifican a una sola interfaz. Un paquete enviado a una dirección unicast es entregado sólo a la interfaz identificada con dicha dirección. Es el equivalente a las direcciones IPv4 actuales. [RFC 2373][RFC 2374]
- **Anycast**, que identifican a un conjunto de interfaces (típicamente pertenecen a diferentes nodos). Un paquete enviado a una dirección anycast, será entregado a una (cualquiera) de las interfaces identificadas con la dirección del conjunto al cual pertenece esa dirección anycast (la más próxima, de acuerdo a las medidas de distancia del protocolo de encaminamiento). Nos permite crear, por ejemplo, ámbitos de redundancia, de forma que varias máquinas puedan ocuparse del mismo tráfico según una secuencia determinada (por el routing), si la primera "cae". [RFC 2526]
- **Multicast** que identifican un grupo de interfaces (por lo general pertenecientes a diferentes nodos). Cuando un paquete es enviado a una dirección multicast es entregado a todas las interfaces del grupo identificadas con esa dirección. En el IPv6 no existen direcciones broadcast, su funcionalidad ha sido mejorada por las direcciones multicast. [RFC 2375]

Vemos que desaparece la dirección Broadcast utilizada en IPv4 ya que lo educado es el empleo de Multicast, ya que sólo tiene que recibirlo la máquina que quiere recibirlo, el resto no deben recibir esos paquetes.

La estructura de las direcciones es la siguiente:

Dirección IPv6 = Prefijo + Id. de Interfaz

Figura 4. Estructura de las direcciones IPv6

La idea a la hora de diseñar el modelo de direcciones es separar en la dirección la parte que identifica “quién eres” y la parte que indica “dónde estás conectado”. Teniendo esto en cuenta tenemos que:

- **Prefijo:** depende de la topología de la red e indica “dónde estás conectado”.
- **Identificador de interfaz:** identifica a un nodo o lo que es lo mismo, “Quién eres”.

Un ejemplo podría ser:



Figura 5. Ejemplo de dirección IPv6

De acuerdo con esta estructura se podría pensar que la dirección del identificador de interfaz estaría muy sobredimensionado ya que se reservan 64 bits para definirlo (no tiene sentido una red en la que haya 2^{64} equipos). Al reservar tantos bits se hizo pensando en:

- Facilitar la autoconfiguración a partir de direcciones MAC más algo más, ya que se piensa que en un futuro la dirección MAC de 48 bits se quedará pequeña. La máquina al arrancar se construye la dirección de la misma.
- Dificultar los ataques sistemáticos (“address/port scanning”) ya que para escanear todas las máquinas posibles de una red (2^{64} equipos) se requiere mucho tiempo.

Además, en previsión de garantizar la privacidad de las máquinas cuando naveguen por Internet, se ha previsto una extensión a la autoconfiguración que permite configurar las máquinas (la parte de identificador de interfaz) con números aleatorios que varíen con el tiempo. De esta forma, aunque conozcan la dirección MAC de una máquina, no pueden saber la dirección IP de la misma. De esta forma se consigue:

- Garantizar la privacidad al navegar por Internet (se genera la dirección local de forma aleatoria y se cambia cada cierto tiempo).
- Para conexiones locales se hace uso de la dirección MAC para generar la dirección local.

2.4 Estrategias de transición a IPv6 (RFC1933)

La clave para la transición es la compatibilidad con la base instalada de dispositivos IPv4. Esta afirmación define un conjunto de mecanismos que los hosts y routers IPv6 pueden implementar para ser compatibles con host y routers IPv4.

Estos mecanismos permitirán usar infraestructuras IPv4 para IPv6 y viceversa, dado que se prevé que su uso será prolongado, e incluso indefinido en muchas ocasiones.

Los mecanismos de transición son un conjunto de mecanismos y de protocolos implementados en hosts y routers, junto con algunas guías operativas de direccionamiento designadas para hacer la transición de Internet al IPv6 con la menor interrupción posible.

Dichos mecanismos están diseñados para ser usados por hosts y routers IPv6 que necesitan inter-operar con hosts IPv4 y utilizar infraestructuras de enrutamiento IPv4. No obstante, IPv6 también puede ser usado en ambientes donde no se requiere interoperabilidad con IPv4. Nodos diseñados para esos ambientes no necesitan usar ni implementar estos mecanismos.

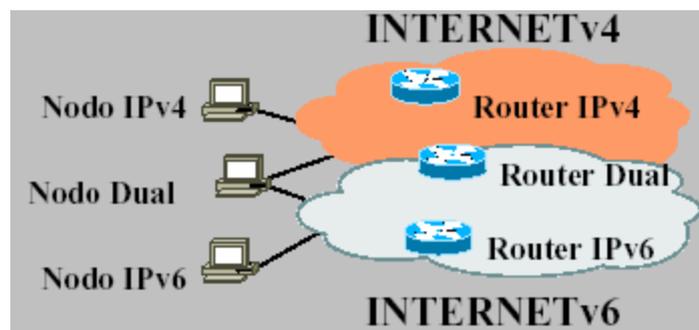


Figura 6. Transición de IPv6 a IPv4

La transición es muy compleja ya que se deben actualizar routers, hosts, aplicaciones, gestión y operación, formación del personal, etc. Conscientes de la complejidad de la transición, IPv6 se ha diseñado pensando en la transición.

Estos mecanismos son los siguientes:

- **Nodos duales o doble pila IPv6 -IPv4.**
- **Túneles.**
- **Traductores de protocolos.**

De estos tres mecanismos en este Trabajo Fin de Máster utilizaremos dos de ellos: la pila dual de protocolos y los traductores de protocolos. Veamos con más detalle cada uno de estos mecanismos.

2.4.1 Nodos duales o doble pila IPv4- IPv6 (Dual Stack)

El camino más lógico y evidente de transición es el uso simultáneo de ambos protocolos, en pilas separadas. Los dispositivos con ambos protocolos también se denominan “nodos IPv6/IPv4”.



Figura 7. Pila dual IPv4 e IPv6

Estos nodos se caracterizan por:

- El dispositivo tendrá una dirección en cada pila. Se pueden utilizar direcciones IPv4 e IPv6 relacionadas o no, y se pueden utilizar mecanismos manuales o automáticos para la asignación de las direcciones (cada una correspondiente al protocolo en cuestión).
- Sólo se duplica el nivel IP, no la pila completa.
- Se comunican utilizando:
 - IPv6 con nodos ya migrados.
 - IPv4 con nodos no migrados.

De esta forma, un dispositivo con ambas pilas pueden recibir y enviar tráfico a nodos que sólo soportan uno de los dos protocolos (nodos sólo IPv4 o sólo IPv6). Estos nodos tienen la habilidad de enviar y recibir paquetes IPv6 e IPv4, pudiendo así inter-operar directamente con nodos IPv4 usando paquetes IPv4, y también operar con nodos IPv6 usando paquetes IPv6.

- Incluyen bibliotecas de DNS (resolver) capaces de tratar con registros A y AAAA. El DNS es quién conoce qué nodos han migrado (registros AAAA disponibles).
- Las aplicaciones deben ser conscientes de la dualidad:
 - Los clientes deben elegir entre v4 o v6 a la hora de conectarse.
 - Los servidores de DNS deben escuchar en v4 y v6.
- Ante una consulta al DNS sobre un nombre, el DNS podrá devolver la dirección IPv4 (registro A), la dirección IPv6 (registro AAAA), o ambas.
- En las redes pueden coexistir paquetes IPv4 e IPv6 ya que los protocolos de nivel inferior en los paquetes existe un campo que identifica al protocolo que transporta:
 - 0x0800 si el protocolo es IPv4.
 - 0x86dd si el protocolo es IPv6.
- Habrá tablas de encaminamiento para IPv4 y para IPv6.
- Hay muchos sistemas como Linux o Windows que ya lo soportan.

Este funcionamiento básico del funcionamiento de la pila dual de protocolos que hemos explicado se representa en la siguiente figura:

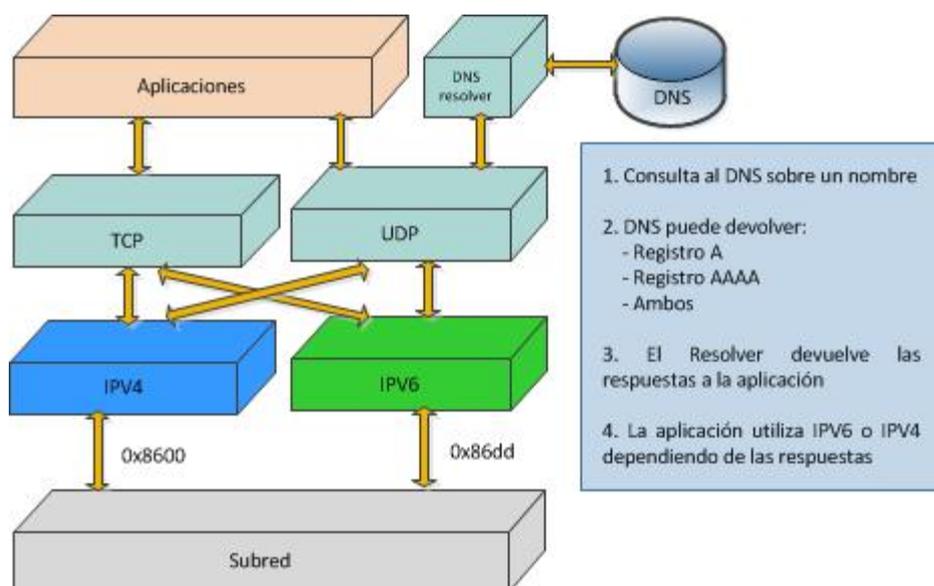


Figura 8. Funcionamiento de la Pila Dual

Se pueden emplear la dirección IPv4 (32 bits), anteponiéndole 80 bits con valor cero y 16 bits con valor 1, para crear una dirección IPv6 “mapeada desde IPv4”.

Algunas recomendaciones a tener en cuenta en los nodos duales son:

- No es conveniente registrar en el DNS las direcciones IPv6 de un sistema hasta que el mismo esté configurado y accesible mediante IPv6 ya que provoca retardos debidos a “timeouts” de TCP al intentar las conexiones IPv6.
- No es conveniente activar IPv6 en un sistema si no tiene conectividad IPv6 ya que produce retardos e incluso problemas de accesibilidad a servidores IPv6.

En este Trabajo Fin de Máster crearemos un escenario con varios servidores configurados con Pila Dual de protocolos para probar tanto el funcionamiento de los mismos como la conectividad entre las distintas redes usando IPv4 e IPv6 indistintamente.

2.4.2 Túneles

Los túneles proporcionan un mecanismo para utilizar las infraestructuras IPv4 mientras la red IPv6 está siendo implantada.

La RFC 2893 define la utilización básica de túneles como mecanismo para transportar paquetes IPv6 sobre redes IPv4.

Los datagramas IPv6 se encapsulan sobre datagramas IPv4 para atravesar redes que aún no han sido migradas.

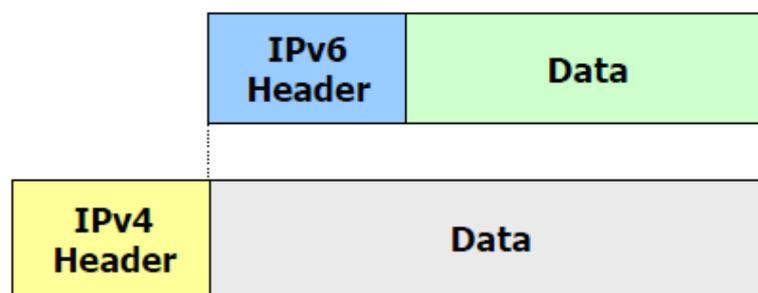


Figura 9. Encapsulación de datagramas

Los túneles se clasifican según el mecanismo por el que el nodo que realiza el encapsulado determina la dirección del nodo extremo del túnel.

- Router-to-router.
- Host-to-Router.
- Router-to-Host.
- Host-to-Host.

En los dos primeros casos (router to router y host to router), el paquete IPv6 es tunelizado a un router. El extremo final de este tipo de túnel, es un router intermedio que debe des-encapsular el paquete IPv6 y reenviarlo a su destino final. En este caso, el extremo final del túnel es distinto del destino final del paquete, por lo que la dirección en el paquete IPv6 no proporciona la dirección IPv4 del extremo final del túnel. La dirección del extremo final del túnel ha de ser determinada a través de información de configuración en el nodo que realiza el túnel. Es lo que se denomina “túnel configurado”, describiendo aquel tipo de túnel donde el extremo final del túnel es explícitamente configurado.

En los otros dos casos (host to host y router to host), el paquete IPv6 es tunelizado, durante todo el recorrido, a su nodo destino. El extremo final del túnel es el nodo destino del paquete, y por tanto, la dirección IPv4 está contenida en la dirección IPv6. Este caso se denomina “túnel automático”.



Figura 10. Túnel IPv6 encapsulado en IPv4

2.4.3 Traductores de protocolos

Los traductores de protocolos se encargan de traducir datagramas IPv6 a IPv4 y viceversa. Permiten la comunicación entre sistemas sólo IPv4 y sistemas sólo IPv6. Traducen las cabeceras de los paquetes entre IPv4 e IPv6 (sólo los campos comunes).

El objetivo de las técnicas de traducción de protocolos es proveer de rutas transparentes a los nodos en las redes de IPv6 para comunicarlos con los nodos de redes IPv4 y viceversa.

Algunos mecanismos de traducción son NAT-PT, TCP-UDP Relay, Socks-based Gateway, Bump-in-the-Stack, Bump-in-the-API, etc...

El mecanismo de traducción que utilizaremos en este Trabajo Fin de Máster será NAT64. Se trata de un mecanismo para permitir a los clientes IPv6 comunicarse con servidores IPv4. El servidor NAT64 es el punto final de al menos una dirección IPv4 y un segmento de red de 32 bits IPv6.

El servidor NAT crea un mapeado NAT entre la IPv6 y la dirección IPv4 permitiendo su comunicación. Para ello se hace necesario el uso de sistemas DNS64.

Un sistema DNS64 se utiliza para compatibilizar ambos sistemas de resolución de nombres. Si el usuario utiliza IPv6, cuando acceda a una web desde el navegador solicitará un registro IPv6 (AAAA). Es decir, a qué IPv6 corresponde un nombre de dominio, en caso de que esa web todavía no se haya adaptado al nuevo protocolo, DNS64 hará una petición IPv4 automáticamente (registro A), a la que sí responderán los servidores DNS, para luego traducir esa dirección a IPv6 y pasarla al usuario. La traducción a una dirección IPv6 consiste en embeber la dirección IPv4 en la dirección IPv6 (por ejemplo 64:ff9b::/96, véase [RFC 6052](#), [RFC 6146](#)). La dirección IPv6 destino se construye utilizando el rango anterior de 96 bits más los 32 bits de la dirección IPv4 con la que desea comunicarse, enviando los paquetes a la dirección resultante.

Por otro lado, NAT64 traduce las direcciones IPv6 del lado del usuario en IPv4 para acceder a Internet en caso de hacerse necesario. Utiliza las mismas tablas que el NAT tradicional para que desde una red privada se acceda a Internet, aunque con adaptación a IPv6. Después de resolver el nombre en un acceso web por DNS64, se hace la petición web a esa dirección IPv6 a través de NAT64, que ya se encargará de volver a traducir a IPv4 para acceder finalmente al servidor.

El principio de funcionamiento puede pensarse como un router con al menos dos interfaces, una de ellas está conectada a la red IPv4 y la otra a la IPv6. El router lleva a cabo las traducciones necesarias para transferir paquetes de IPv6 a IPv4 y viceversa

(teniendo en cuenta que esta traducción no es simétrica ya que el espacio de direcciones de IPv6 es de 2^{128} frente a 2^{32} del IPv4).

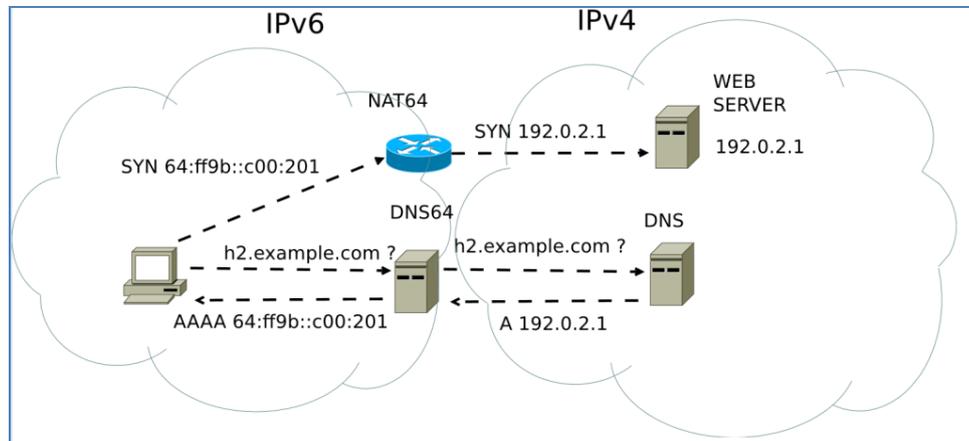


Figura 11. Funcionamiento del Traductor NAT64

3 Virtualización y VNX

La reducción de precios en el hardware y la presencia de productos software de virtualización cada vez más asequibles han conseguido que la virtualización, antes reservada a grandes compañías, pueda hoy ser utilizada casi por cualquier persona u organización. Bien utilizada, la virtualización puede poner a nuestra disposición más opciones en el despliegue de sistemas, costes menores y un mayor control sobre nuestra estructura.

3.1 Virtualización

Desde un punto de vista genérico, puede definirse la virtualización como una técnica (normalmente con un software asociado) que permite encapsular una unidad de proceso (ya sea un programa, un sistema operativo o incluso un equipo completo, dependiendo de a qué profundidad se sitúe el nivel de virtualización) para su ejecución dentro de un entorno en un equipo anfitrión que emula el entorno real de forma transparente.

Esto implica que, en disposición de una máquina lo suficientemente potente que actúe como equipo anfitrión, es posible ejecutar simultáneamente un sistema de “máquinas virtuales” que se comporten de forma equivalente al mismo sistema implementado con máquinas reales. El grado de similitud entre la implementación

virtual del sistema y la implementación con equipos reales puede tomarse como una medida de la bondad de la técnica de virtualización empleada: lo deseable es que el sistema emulado se comporte lo más transparentemente posible, idealmente exactamente igual que el sistema real.

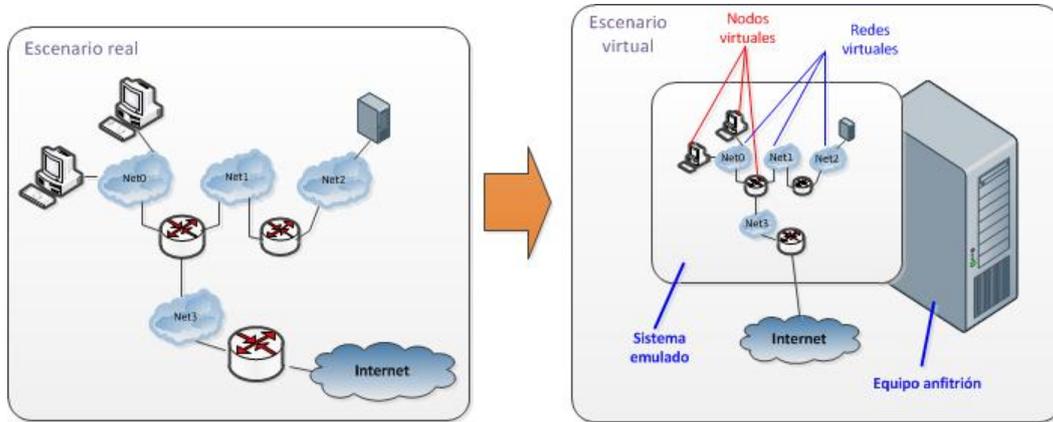


Figura 12. Sistema Real (izquierda) y sistema virtualizado (derecha)

Las ventajas de disponer de sistemas virtualizados frente a sistemas físicos son las siguientes:

- **Ahorro de costes:** Sólo es necesario adquirir una máquina física que funcione como máquina anfitriona, aunque más potente, y en ella ir creando sistemas en el gestor de máquinas virtuales. También permite ahorro en el coste de mantenimiento y en el de personal, además de ahorrar espacio.
- **Crecimiento más flexible:** Instalar un nuevo servidor es mucho más sencillo y rápido frente a hacerlo con un servidor físico.
- **Administración simplificada:** Desde la consola del gestor de máquinas virtuales podemos aumentar o reducir los recursos para una determinada máquina, reiniciarla, instalar parches o simplemente borrarla en caso de problemas.
- **Centralización de tareas de mantenimiento:** Podemos realizar copias de seguridad de todas las máquinas virtuales a la vez, programar actualizaciones y otras actividades desde el gestor de máquinas virtuales. También podemos centralizar otras funciones.
- **Disminución de los tiempos de parada:** Una ventaja importante, solucionar problemas o realizar copias de seguridad son tareas que se realizan en mucho menos tiempo. Por ejemplo, se puede clonar una máquina y seguir dando servicio mientras se realiza mantenimiento de la máquina virtual de producción como puede ser una actualización.

- **Mejor gestión de recursos:** Se puede aumentar la memoria o almacenamiento de la máquina huésped para aumentar los recursos de todas las máquinas virtuales a la vez, por lo que se aprovecha mucho mejor las inversiones en hardware.
- **Balanceo de recursos:** Es posible asignar un grupo de servidores físicos para que proporcionen recursos a las máquinas virtuales y asignar una aplicación que haga un balanceo de los mismos, otorgando más memoria, recursos de la CPU, almacenamiento o ancho de banda de la red a la máquina virtual que lo necesite.

Sin embargo, es difícil conseguir una transparencia completa usando virtualización. Esto se debe a que la virtualización introduce un nivel de proceso adicional (el que traduce las llamadas al sistema de la máquina virtual al sistema anfitrión) que supone un overhead debido al consumo de recursos. Para conseguir un rendimiento equivalente al del sistema real es necesario utilizar hardware de mayor potencia en la máquina anfitriona. En todo caso, esto es algo que depende de lo eficiente que sea la técnica de virtualización que se emplee.

Si bien la teoría en la que se basan las técnicas de virtualización modernas es bastante antigua [1], el auge de este tipo de soluciones no se ha producido hasta hace relativamente poco tiempo, motivadas principalmente por la baja relación *potencia del hardware/precio* a la que hemos llegado en nuestros días. Algunas de soluciones para ejecutar máquinas virtuales sobre un equipo anfitrión son: Xen [2], VMware [3], UML (User Mode Linux) [4].

Combinando las máquinas virtuales con el uso de redes virtuales emuladas en la máquina anfitrión es posible crear Escenarios de Red Virtuales incluso con conexiones externas. Algunas herramientas de gestión de máquinas virtuales son: GNS3 [5], Netkit [6], Marionnet [7], VNX/VNUML [8], etc.

3.2 VNX/VNUML

VNX/VNUML es una herramienta de software libre formada por dos componentes: un lenguaje simple y descriptivo basado en XML (eXtended Markup Language [9]) que permite al usuario definir el sistema emulado en un fichero de texto; y un parser (escrito en Perl [10]) de ese lenguaje que se encarga de procesar el fichero, construir la simulación y gestionarla, ocultando los detalles complejos al usuario.

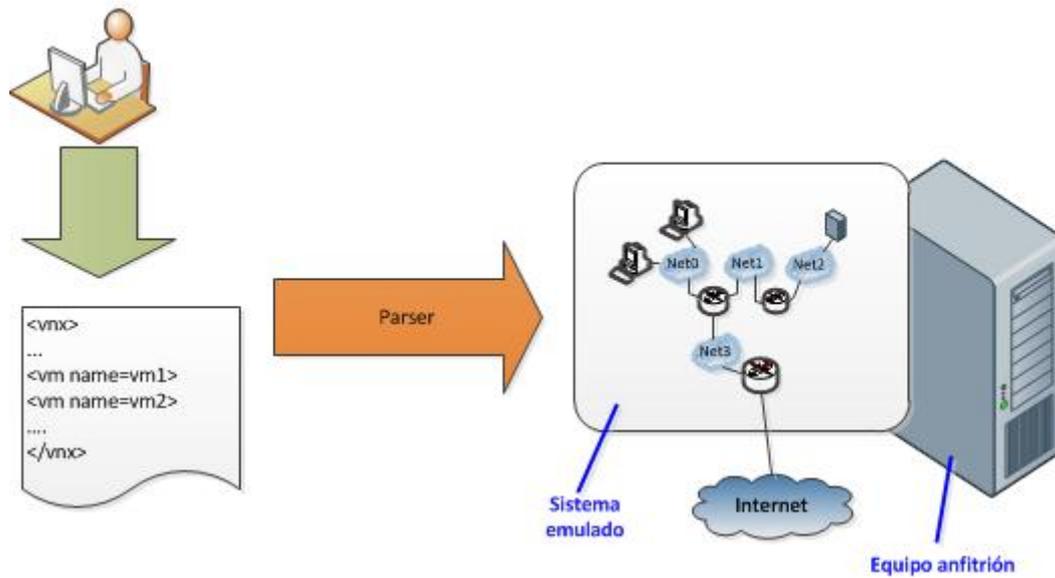


Figura 13. Funcionamiento básico de VNX

XML es el lenguaje base para la descripción de escenarios de simulación. Propuesto por el W3C en 1998 tras el auge del lenguaje HTML (Hypertext Markup Language) en la web, XML surge como un estándar abierto para el intercambio de información por Internet basado en etiquetas de texto. Comparado con otras alternativas, esta orientación a etiquetas hace que el procesado del lenguaje no sea muy eficiente, pero sí muy sencillo y muy intuitivo para el usuario, características por las cuales se ha elegido en VNX para representar las descripciones de escenario. XML tiene a su vez un conjunto de estándares asociados, de los que se hace uso en VNX: DTD (Document Type Definition), para la definición de la estructura de etiquetas; y DOM (Document Object Model), utilizado por el parser para procesar los datos de la simulación.

Como sistema de virtualización VNUML (Virtual Networks User Mode Linux) utiliza UML. UML es una modificación de las fuentes del núcleo de Linux [11] que permiten su ejecución como proceso de usuario encima del núcleo convencional de Linux (el que ejecuta la máquina anfitriona). Cada uno de los procesos UML tiene asociados sus propios recursos (espacio de memoria, procesos, sistemas de ficheros, dispositivos de red, etc.) y, en definitiva, constituye una máquina virtual dentro de la cual otros procesos se ejecutan. La funcionalidad del núcleo UML es exactamente la misma que la de un núcleo convencional de Linux (de hecho, el proceso de compilación es análogo), por lo que la transparencia es (salvo por la pérdida de rendimiento debido al overhead de virtualización) total.

UML no sólo proporciona la manera de crear máquinas virtuales, sino que existen mecanismos para la interconexión entre ellas mediante redes virtuales. Desde el punto de vista de la máquina virtual, se utiliza una interfaz de red de forma transparente. En

el entorno anfitrión, la red virtual interconecta los distintos terminales utilizando el módulo de túneles del núcleo. Es de destacar como la interconexión se produce en modo bridge, a nivel 2, y, a todos los efectos, es como si las máquinas interconectadas a una misma red virtual estuvieran físicamente conectadas a un mismo segmento. También es posible que la máquina anfitriona intervenga en la simulación o interconexión de máquinas virtuales con equipos externos a través del interfaz físico (lo cual, es importante a la hora de integrar sistemas no-Linux en la simulación), de forma completamente transparente o a través de VLANs (Virtual Local Area Network [12]).

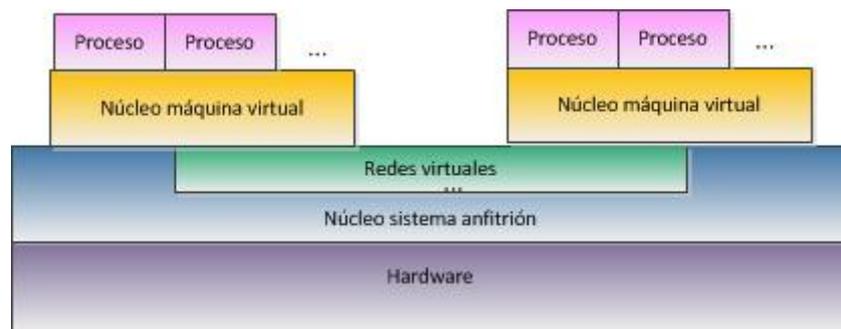


Figura 14. Virtualización con VNX

VNX/VNUML también proporciona facilidades en la gestión de las máquinas virtuales. En todas ellas existe una interfaz de red reservada, que permite una conexión directa con la máquina anfitriona, con el objetivo de realizar operaciones de gestión a través de la misma (por ejemplo, a través de SSH) [13]. Existen otros mecanismos de gestión de las máquinas virtuales, como el uso de consolas directas en las mismas (por ejemplo, un xterm) o a bajo nivel a través de un socket UNIX que comunica directamente con el núcleo UML.

VNX se distribuye con licencia libre GPL y con vocación de herramienta de uso público para investigadores y docentes. Está basada en VNUML del que mantiene muchas características e incorpora algunas nuevas funcionalidades como:

- Integración de libvirt (acceso estándar a virtualización de Linux).
- Autoconfiguración para Windows XP, Windows 7, Linux (Ubuntu 9.10/10.04/11.04), FreeBSD (8.1) y Fedora 14.
- Integración Dynamips (CISCO).
- Integración Olive (Juniper).
- Funcionalidad de gestión individual de máquinas.

3.3 Funcionamiento de VNX

El funcionamiento de VNX/VNUML se refleja en los pasos indicados en la siguiente figura:

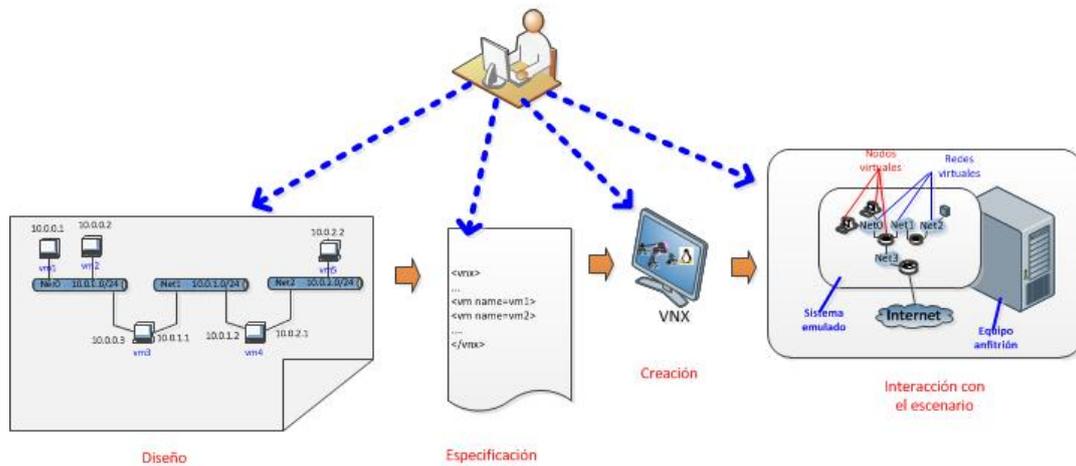


Figura 15. Funcionamiento de VNX

En primer lugar, el usuario diseña el escenario que pretende simular, de manera off-line. A continuación especifica el escenario en lenguaje VNX/VNUML usando XML. Una vez escrito, VNX procesa la especificación y crea el escenario virtual que ya podemos arrancar. En este escenario podemos ejecutar comandos en las máquinas virtuales a través de VNX o directamente sobre las mismas.

Podemos concluir que la utilización de técnicas de virtualización y, en particular, herramientas de virtualización de escenarios como VNX en infraestructuras de experimentación permite:

- Un ahorro de costes, tanto de equipamiento como esfuerzo invertido en su gestión y configuración.
- Mejor compartición de infraestructuras de experimentación.
- Crear y reutilizar escenarios complejos mediante un esfuerzo razonable.
- Concentrar el esfuerzo en el sistema o servicio a probar y no en la infraestructura de pruebas.

4 Migración a IPv6 en una escuela de la UPM (EIAE)

4.1 Introducción

Son muchas las motivaciones que pueden dar lugar a implantar el protocolo IPv6 en las redes de la EIAE. Anteriormente se han descrito algunas de las ventajas técnicas que ofrece IPv6.

La implantación del protocolo IPv6 se debe, además de las ventajas técnicas, a la demanda producida por parte del personal investigador de la EIAE como soporte para muchos de los proyectos de investigación en los que trabajan.

Pero la migración es un proceso largo y costoso que comienza con la implantación del protocolo de red en la infraestructura de comunicaciones de la EIAE, continuando con la modificación de los servicios ofrecidos actualmente en ella y finalizará con la instalación del protocolo en todos los dispositivos conectados a su infraestructura de comunicaciones.

La implantación del protocolo de red IPv6 lleva consigo una serie de requisitos que deben cumplirse para poderse llevar a cabo de forma eficaz la migración al mismo. Entre los muchos requisitos, podemos destacar los siguientes:

- Acceso de todos los usuarios a la nueva red.
- Acceso a los servicios de información de IPv6.
- Documentación y apoyo técnico a los usuarios.
- Servicio de gestión del direccionamiento IPv6.
- Servicio de gestión de la seguridad.

Todo esto teniendo en cuenta de que la red actual no debe sufrir durante el proceso de migración.

4.2 Modelo de direccionamiento

El modelo de direccionamiento se encarga de realizar una asignación de direcciones IPv6 a cada una de las redes que actualmente existen en la EIAE para dotarlas de conectividad IPv6. El modelo de direccionamiento también tiene en cuenta las posibles necesidades que puede haber en un futuro dentro de la EIAE [14].

El documento RIPE-267 define la política de asignación de direcciones en Europa. En dicho documento, propone asignar un prefijo /32 a los proveedores de servicio de Internet que en menos de dos años asignen a sus usuarios 200 prefijos /48.

Dado que el proveedor de servicios de la UPM es RedIRIS y, de acuerdo con las anteriores recomendaciones, RedIRIS tiene delegado por RIPE el prefijo **2001:0720::/32**.

De acuerdo con el plan de asignación de direcciones establecido por RedIRIS, a cada Universidad se le asigna un prefijo /48. De acuerdo con esto, el prefijo que RedIRIS ha asignado a la UPM es **2001:0720:041c::/48**.

Dado que RedIRIS ha delegado a UPM el prefijo **2001:0720:041c::/48**, esto permite utilizar 16 bits para la Universidad, esto hace un total de 65.536 redes posibles de las que ha asignado a la EIAE el prefijo **2001:0720:041c:4000::/54**.

La EIAE está compuesta por personal docente e investigador, personal de administración y servicios y alumnos organizados en departamentos o adscritos a Servicios Centrales. Las necesidades de direcciones no son iguales en todos los departamentos de la EIAE. Por este motivo es necesario dividir las direcciones IPv6 entre los departamentos según sus necesidades. Así, podemos clasificarlos en tres tipos:

- Un **departamento de tipo 1** es aquel que bien por su mayor número de personal adscrito al mismo, o bien por su número de sistemas, tiene más necesidades de direcciones IPv6. Dentro de este tipo se incluye los Servicios Centrales.
- Un **departamento de tipo 2** es aquel que tiene menos necesidades de direcciones IPv6 que un departamento de tipo 1 pero más que uno de tipo 3.
- Un **departamento de tipo 3** es aquel que menos necesidades de direcciones IPv6. Dentro de este tipo se pueden incluir los proyectos de investigación desarrollados en la EIAE y las cátedras Universidad-Empresa.

De acuerdo con esto y considerando la estructura departamental de la EIAE, en la que la mayor parte de los departamentos son pequeños o no disponen de un gran número de máquinas, el plan de direccionamiento podría ser el representado en la figura 4:

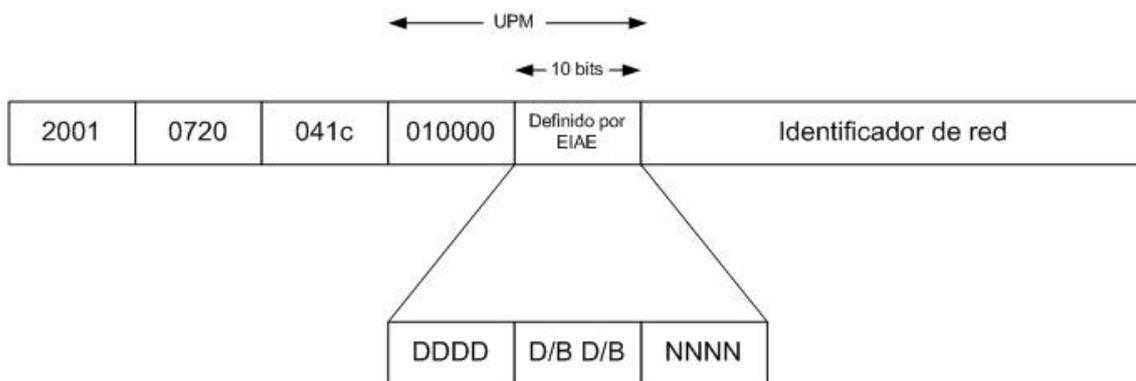


Figura 16. Plan de direccionamiento IPv6 en la EIAE

Donde:

- **D**: representa al identificador de departamento.
- **B**: Representa al identificador de bloque.
- **N**: Representa el identificador de red (dentro de una dirección de bloque asignado a un departamento).

De acuerdo con este plan de direccionamiento, habría:

- Hasta un máximo de 4 departamentos del tipo 1 a los que se les asignará inicialmente dos bloques de 16 redes, quedando otros dos bloques de reserva para futuros usos.
- Hasta un máximo de 16 departamentos del tipo 2 a los que se les asignará inicialmente un bloque de 16 redes, quedando otro bloque de reserva para futuros usos.
- Hasta un máximo de 16 departamentos del tipo 3 a los que se les asignará un bloque de 16 redes.

Tipo de Departamento	Formato de direcciones	Nº Bloques asignados	Nº de bloques de Reserva	Longitud del prefijo
1	DDDDBBNNNN	2	2	/58
2	DDDDDBNNNN	1	1	/59
3	DDDDDDNNNN	1	0	/60

Tabla 1. Reparto por tipos de departamentos en la EIAE

Un posible reparto de las direcciones IPv6 dentro de la EIAE podría ser el siguiente:

Tipo de Departamento	Nombre Departamento	Formato de direcciones	Nº Bloques asignados	Nº de bloques de Reserva	Longitud del prefijo	Ejemplo
1	Departamento-1	0000BNNNN	2	2	/58	2001:720:41c:4000::/58
1	Departamento-2	0001BNNNN	2	2	/58	2001:720:41c:4040::/58
1	Departamento-3	0010BNNNN	2	2	/58	2001:720:41c:4080::/58
1	Departamento-4	0011BNNNN	2	2	/58	2001:720:41c:40c0::/58
2	Departamento-5	01000BNNNN	1	1	/59	2001:720:41c:4100::/59
2	Departamento-6	01001BNNNN	1	1	/59	2001:720:41c:4120::/59
2	Departamento-7	01010BNNNN	1	1	/59	2001:720:41c:4140::/59
2	Departamento-8	01011BNNNN	1	1	/59	2001:720:41c:4160::/59
2	Departamento-9	01100BNNNN	1	1	/59	2001:720:41c:4180::/59
2	Departamento-10	01101BNNNN	1	1	/59	2001:720:41c:41a0::/59
2	Departamento-11	01110BNNNN	1	1	/59	2001:720:41c:41c0::/59
2	Departamento-12	01111BNNNN	1	1	/59	2001:720:41c:41e0::/59
2	Departamento-13	10000BNNNN	1	1	/59	2001:720:41c:4200::/59
2	Departamento-14	10001BNNNN	1	1	/59	2001:720:41c:4220::/59
2	Departamento-15	10010BNNNN	1	1	/59	2001:720:41c:4240::/59
2	Departamento-16	10011BNNNN	1	1	/59	2001:720:41c:4260::/59
2	Departamento-17	10100BNNNN	1	1	/59	2001:720:41c:4280::/59
2	Departamento-18	10101BNNNN	1	1	/59	2001:720:41c:42a0::/59

2	Departamento-19	10110BNNNN	1	1	/59	2001:720:41c:42c0::/59
2	Departamento-20	10111BNNNN	1	1	/59	2001:720:41c:42e0::/59
3	Departamento-21	11000NNNN	1	0	/60	2001:720:41c:4300::/60
3	Departamento-22	110001NNNN	1	0	/60	2001:720:41c:4310::/60
3	Departamento-23	110010NNNN	1	0	/60	2001:720:41c:4320::/60
3	Departamento-24	110011NNNN	1	0	/60	2001:720:41c:4330::/60
3	Departamento-25	110100NNNN	1	0	/60	2001:720:41c:4340::/60
3	Departamento-26	110101NNNN	1	0	/60	2001:720:41c:4350::/60
3	Departamento-27	110110NNNN	1	0	/60	2001:720:41c:4360::/60
3	Departamento-28	110111NNNN	1	0	/60	2001:720:41c:4370::/60
3	Departamento-29	111000NNNN	1	0	/60	2001:720:41c:4380::/60
3	Departamento-30	111001NNNN	1	0	/60	2001:720:41c:4390::/60
3	Departamento-31	111010NNNN	1	0	/60	2001:720:41c:43a0::/60
3	Departamento-32	111011NNNN	1	0	/60	2001:720:41c:43b0::/60
3	Departamento-33	111100NNNN	1	0	/60	2001:720:41c:43c0::/60
3	Departamento-34	111101NNNN	1	0	/60	2001:720:41c:43d0::/60
3	Departamento-35	111110NNNN	1	0	/60	2001:720:41c:43e0::/60
3	Departamento-36	111111NNNN	1	0	/60	2001:720:41c:43f0::/60

Tabla 2. Reparto de prefijos en la EIAE por departamentos

4.3 Estrategia de transición

Una vez que se ha definido el plan de direccionamiento IPv6 en la EIAE se debe comenzar con la transición a la nueva red.

La transición es compleja ya que se deben actualizar routers, hosts, aplicaciones, gestión y operación, formación del personal, etc. Además es un proceso largo ya que durante la migración de IPv4 a IPv6 deben coexistir ambos sistemas por un tiempo que puede llegar a ser indefinido. Como ya hemos comentado la clave para la transición es la compatibilidad con la base instalada de dispositivos IPv4 lo que supone un conjunto de mecanismos que los hosts y routers IPv6 pueden implementar para ser compatibles con host y routers IPv4.

Las consideraciones a tener en cuenta desde el punto de vista de la transición a IPv6 son las siguientes:

- Todos los dispositivos actualmente conectados a la red de la EIAE dispondrán de doble pila IPv4- IPv6. Esta técnica es una solución más fácil de implementar en entornos corporativos como el que queremos tratar.

Ambos protocolos convivirán en el mismo segmento de red (dual-stack network). Esto implica que los dispositivos de red, independientemente de que utilicen IPv4 o IPv6 a nivel de red, comparten la misma red física y por tanto, el mismo dominio de broadcast y la misma vlan. Esto supone que los sistemas operativos de los equipos conectados a las redes van a utilizar ambas pilas de protocolos en paralelo.

Para configurar la dirección IPv6 de los dispositivos conectados a la red de la EIAE se utilizarán métodos de autoconfiguración. En el presente trabajo hemos utilizado el método de autoconfiguración stateful utilizando para ello un servidor DHCPv6 para, además de ofrecer la dirección IPv6, ofrecer otras informaciones como el servidor dns con el que tiene que configurarse, el dominio al que pertenecen, etc.

- Aunque actualmente no es necesario debido a que todos los sistemas conectados a la red de la EIAE disponen de direccionamiento IPv4 público, y una vez desplegado el direccionamiento IPv6, todos los sistemas dispondrían de doble pila, se va a implementar un método de translación de direcciones llamado NAT64.

La estrategia a seguir en el despliegue de IPv6 en la EIAE es la siguiente:

1. Poner la doble pila IPv4-IPv6 en los routers y en los servidores.
2. Configurar los servicios para IPv6.
3. Extender la doble pila a los usuarios finales.
4. Eliminar IPv4 en los sistemas de los usuarios finales cuando todos tengan instalado IPv6.
5. Eliminar IPv4 de los routers y de los servidores.

5 Prototipos

5.1 Simulador

Como hemos comentado en apartados anteriores el simulador elegido para realizar este Trabajo Fin de Máster es VNX/VNUML. En el apartado 3 hemos descrito su funcionamiento básico. Vamos a describir brevemente un poco del lenguaje que utiliza y que usaremos para describir nuestro escenario:

La sintaxis básica es la siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<vnx>
<global>
<version>1.92</version>
<scenario_name>Escenario-Proyecto</scenario_name>
</global>
<net name="Net0" mode="virtual_bridge" external="eth0"/>
<net name="Net1" mode="uml_bridge" />
<net name="Net2" mode="uml_bridge" />
<net name="Net3" mode="uml_bridge" />
<vm> ... </vm>
<vm>... </vm>
...
</vnx>
```

Toda la definición del escenario debe estar englobada entre las etiquetas <vnx> y </vnx>. Las definiciones globales van entre las etiquetas <global> y </global>. Las redes se definen con la etiqueta <net> y las máquinas virtuales se definen entre las

etiquetas `<vm>` y `</vm>`. Dentro de estas etiquetas se definen los parámetros de cada una de las máquinas virtuales:

```
<vmname="cliente-xp1" type="libvirt" subtype="kvm" os="windows">
<filesystemtype="cow"/>/usr/share/vnx/filesystems/root_fs_winxp</filesystem>
<mem>256M</mem>
<if id="1" net="Net2">
<ipv4>10.0.2.2/24</ipv4>
</if>
<route type="ipv4" gw="10.0.2.1">10.0.2.1/24</route>
</vm>
<vmname="srv-www" type="libvirt" subtype="kvm" os="linux">
<filesystemtype="cow"/>/usr/share/vnx/filesystems/root_fs_ubuntu</filesystem>
<mem>256M</mem>
<if id="1" net="Net3">
<ipv4>10.0.3.2/24</ipv4>
</if>
<route type="ipv4" gw="10.0.0.1">10.0.0.1/24</route>
</vm>
```

La etiqueta `<filesystem>` define el sistema de ficheros, la etiqueta `<mem>` define la cantidad de memoria asignada a la máquina virtual, la etiqueta `<if>` nos permite configurar una dirección IP y la etiqueta `<route>` poner una ruta por defecto. A modo de ejemplo algunos de los comandos más usados en VNX son:

- Arranque del escenario:

```
# vnx -f escenario.xml --create
```

- Acceso a consolas:

```
# vnx -f escenario.xml --console -M vm1
```

- Ejecución de comandos:

```
# vnx -f escenario.xml --execute start
```

- Re-arranque de una máquina virtual:

```
# vnx -f escenario.xml --reboot -M vm1
```

- Parada del escenario:

```
# vnx -f escenario.xml --shutdown
```

```
# vnx -f escenario.xml --destroy
```

5.2 Descripción del escenario

El escenario creado pretende simular la red y los posibles servicios ofrecidos por una escuela de la UPM y probar sobre ella diferentes servicios tanto con IPv4 como con IPv6.

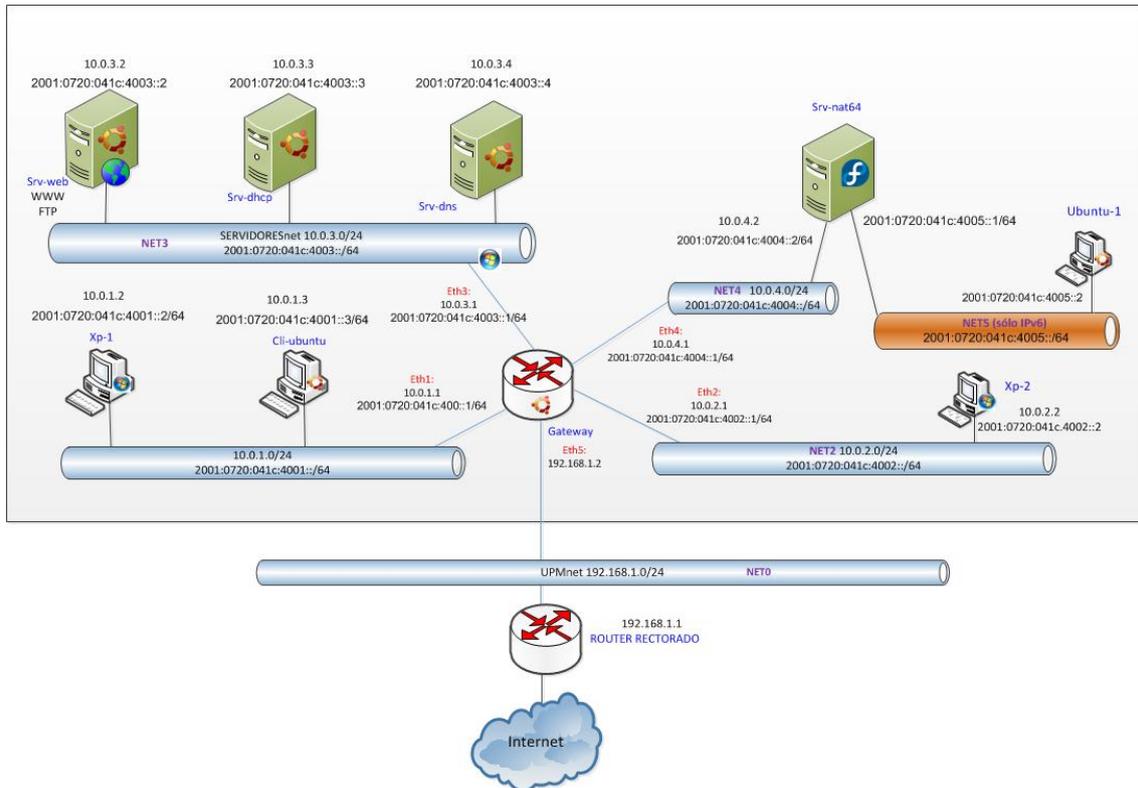


Figura 17. Escenario de simulación

Los sistemas utilizados en esta simulación son los siguientes:

- **Gateway:** es el router principal de la escuela.
- **Srv-web:** es el servidor web y ftp de la escuela.
- **Srv-dns:** es el servidor dns de la escuela.
- **Srv-dhcp:** es el servidor dhcp de la escuela.
- **Srv-nat64:** es el servidor NAT64 de la escuela.
- **Xp-1:** es un cliente con sistema operativo Windows XP.
- **Cli-ubuntu:** es un cliente con sistema operativo Ubuntu.
- **Xp-2:** es un cliente con sistema operativo Windows XP.
- **Ubuntu-1:** es un cliente con sistema operativo Ubuntu.

Todos los equipos están conectados a las siguientes redes:

Nombre	Dirección IPv4	Dirección IPv6	Descripción
NET0	192.168.1.0/24	-	Red de conexión a la red de UPM
NET1	10.0.1.0/24	2001:720:41c:4001::/64	Red de Clientes
NET2	10.0.2.0/24	2001:720:41c:4002::/64	Red de Clientes
NET3	10.0.3.0/24	2001:720:41c:4003::/64	Red de Servidores
NET4	10.0.4.0/24	2001:720:41c:4004::/64	Red para la translación IPv4- IPv6
NET5	-	2001:720:41c:4005::/64	Red con sistemas configurados sólo con IPv6

Tabla 3. Redes utilizadas en la simulación

Las características de todos los sistemas utilizados en la virtualización es la siguiente:

Nombre	Dirección IPv4	Dirección IPv6	Sistema Operativo
Gateway	192.168.1.2	2001:720:41c:4001::1	Ubuntu 10.10
	10.0.1.1	2001:720:41c:4002::1	
	10.0.2.1	2001:720:41c:4003::1	
	10.0.3.1	2001:720:41c:4004::1	
	10.0.4.1		
Srv-web	10.0.3.2	2001:720:41c:4003::2	Ubuntu 10.10
Srv-dhcp	10.0.3.3	2001:720:41c:4003::3	Ubuntu 10.10
Srv-dns	10.0.3.4	2001:720:41c:4003::4	Ubuntu 10.10

Srv-nat64	10.0.4.2	2001:720:41c:4004::2 2001:720:41c:4005::1	Fedora 14
Xp-1	10.0.1.2	2001:720:41c:4001::2	Windows XP SP3
Xp-2	10.0.2.2	2001:720:41c:4002::2	Windows XP SP3
Cliente-ubuntu	10.0.1.3	2001:720:41c:4001::3	Ubuntu 10.10
Ubuntu-1	-	2001:720:41c:4005::2	Ubuntu 10.10

Tabla 4. Sistemas utilizados en la simulación

El fichero xml que permite crear este escenario es el siguiente:

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
Trabajo Fin de Master
AUTORES: Rafael García García
          Gloria Martín Martín
-->
<vnx xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:noNamespaceSchemaLocation="/usr/share/xml/vnx/vnx-1.95.xsd">
  <global>
    <version>1.92</version>
    <scenario_name>Escenario</scenario_name>
    <automac/>
    <vm_mgmt type="none" />
    <vm_defaults exec_mode="mconsole">
      <console id="0" display="yes"/>
      <console id="1" display="no"/>
    </vm_defaults>
  </global>

  <net name="Net0" mode="virtual_bridge" external="eth0"/>
  <net name="Net1" mode="virtual_bridge" />
  <net name="Net2" mode="virtual_bridge" />
  <net name="Net3" mode="virtual_bridge" />

```

```

<net name="Net4" mode="virtual_bridge" />
<net name="Net5" mode="virtual_bridge" />
<!-- NODOS -->

<vm name="Xp-1" type="libvirt" subtype="kvm" os="windows">
  <filesystem type="cow">/usr/share/vnx/filesystems/root_fs_winxp</filesystem>
  <mem>256M</mem>
  <if id="1" net="Net1">
  </if>
</vm>

<vm name="Cli-ubuntu" type="libvirt" subtype="kvm" os="linux">
  <filesystem type="cow">/usr/share/vnx/filesystems/root_fs_ubuntu</filesystem>
  <mem>256M</mem>
  <if id="1" net="Net1">
  </if>
  <filetree seq="redes" root="/root/cli-ubuntu">/root/cli-ubuntu/</filetree>
  <exec seq="red" type="verbatim" mode="system">cp /root/cli-ubuntu/interfaces.
/etc/network/interfaces</exec>
  <exec seq="reinicio-red" type="verbatim" mode="system">/etc/init.d/networking
restart</exec>
</vm>

<vm name="Ubuntu-1" type="libvirt" subtype="kvm" os="linux">
  <filesystem type="cow">/usr/share/vnx/filesystems/root_fs_ubuntu</filesystem>
  <mem>256M</mem>
  <if id="1" net="Net5">
    <ipv6>2001:0720:41c:4005::2/64</ipv6>
  </if>
  <route type="ipv6" gw="2001:0720:41c:4005::1">2001:0720:41c:4005::1/64</route>
  <exec seq="rutadefecto" type="verbatim" mode="system">route -A inet6 add default gw
2001:720:41c:4005::1</exec>
</vm>

<vm name="Xp-2" type="libvirt" subtype="kvm" os="windows">
  <filesystem type="cow">/usr/share/vnx/filesystems/root_fs_winxp</filesystem>
  <mem>256M</mem>
  <if id="1" net="Net2">
  </if>
</vm>

```

```

<vm name="Srv-web" type="libvirt" subtype="kvm" os="linux">
  <filesystem
type="direct">/usr/share/vnx/filesystems/root_fs_ubuntu_www.qcow2</filesystem>
  <mem>256M</mem>
  <if id="1" net="Net3">
    <ipv4>10.0.3.2/24</ipv4>
    <ipv6>2001:0720:41c:4003::2/64</ipv6>
  </if>
  <route type="ipv4" gw="10.0.3.1">10.0.3.1/24</route>
  <route type="ipv6" gw="2001:0720:41c:4003::1">2001:0720:41c:4003::1/64</route>
  <filetree seq="apache" root="/root/apache">/root/apache/</filetree>
  <filetree seq="paginas" root="/root/ipv6">/root/ipv6/</filetree>
  <filetree seq="ftp" root="/root/ftp">/root/ftp/</filetree>
  <exec seq="copia-1" type="verbatim" mode="system">cp /root/apache/default.
/etc/apache2/sites-available/default</exec>
  <exec seq="copia-2" type="verbatim" mode="system">cp /root/apache/ports.conf
/etc/apache2/ports.conf</exec>
  <exec seq="copia-3" type="verbatim" mode="system">cp /root/apache/hosts.
/etc/hosts</exec>
  <exec seq="copia-4" type="verbatim" mode="system">cp /root/apache/default_ssl.
/etc/apache2/sites-available/default-ssl</exec>
  <exec seq="copia-5" type="verbatim" mode="system">a2enmod ssl</exec>
  <exec seq="copia-6" type="verbatim" mode="system">a2ensite default-ssl</exec>
  <exec seq="copia-7" type="verbatim" mode="system">cp -a /root/ipv6/ /var/www/</exec>
  <exec seq="copia-8" type="verbatim" mode="system">chmod -R o+rx
/var/www/ipv6</exec>
  <exec seq="copia-9" type="verbatim" mode="system">/etc/init.d/apache2 restart</exec>
  <exec seq="copia-ftp" type="verbatim" mode="system">cp /root/ftp/vsftpd.conf
/etc/vsftpd.conf</exec>
  <exec seq="reinicio-ftp" type="verbatim" mode="system">/etc/init.d/vsftpd restart</exec>
  <exec seq="dns" type="verbatim" mode="system">echo "nameserver 10.0.3.4" >
/etc/resolv.conf</exec>
</vm>

```

```

<vm name="Srv-dhcp" type="libvirt" subtype="kvm" os="linux">
  <filesystem
type="direct">/usr/share/vnx/filesystems/root_fs_ubuntu_dhcp.qcow2</filesystem>
  <mem>256M</mem>
  <if id="1" net="Net3">
    <ipv4>10.0.3.3/24</ipv4>

```

```

    <ipv6>2001:0720:41c:4003::3/64</ipv6>
  </if>
  <route type="ipv4" gw="10.0.3.1">10.0.3.1/24</route>
  <route type="ipv6" gw="2001:0720:41c:4003::1">2001:0720:41c:4003::1/64</route>
  <exec seq="dns" type="verbatim" mode="system">echo "nameserver 10.0.3.4" >
/etc/resolv.conf</exec>
</vm>

<vm name="Srv-dns" type="libvirt" subtype="kvm" os="linux">
  <filesystem
type="direct">/usr/share/vnx/filesystems/root_fs_ubuntu_dns.qcow2</filesystem>
  <mem>256M</mem>
  <if id="1" net="Net3">
    <ipv4>10.0.3.4/24</ipv4>
    <ipv6>2001:0720:41c:4003::4/64</ipv6>
  </if>
  <route type="ipv4" gw="10.0.3.1">10.0.3.1/24</route>
  <route type="ipv6" gw="2001:0720:41c:4003::1">2001:0720:41c:4003::1/64</route>
  <exec seq="dns" type="verbatim" mode="system">echo "nameserver 10.0.3.4" >
/etc/resolv.conf</exec>
</vm>

<vm name="Srv-nat64" type="libvirt" subtype="kvm" os="linux">
  <filesystem
type="direct">/usr/share/vnx/filesystems/root_fs_fedora-14-v01-
nat64.qcow2</filesystem>
  <mem>256M</mem>
  <if id="1" net="Net4">
    <ipv4>10.0.4.2/24</ipv4>
    <ipv6>2001:0720:41c:4004::2/64</ipv6>
  </if>
  <if id="2" net="Net5">
    <ipv6>2001:0720:41c:4005::1/64</ipv6>
  </if>
  <route type="ipv4" gw="10.0.4.1">10.0.4.1/24</route>
  <route type="ipv6" gw="2001:0720:41c:4004::1">2001:0720:41c:4004::1/64</route>
  <exec seq="dns" type="verbatim" mode="system">echo "nameserver 10.0.4.2" >
/etc/resolv.conf</exec>
  <exec seq="rutanat64" type="verbatim" mode="system">route -A inet6 add ::0/0 gw
2001:720:41c:4004::1</exec>
</vm>

```

```

<vm name="Gateway" type="libvirt" subtype="kvm" os="linux">
  <filesystem
type="direct">/usr/share/vnx/filesystems/root_fs_ubuntu_router.qcow2</filesystem>
  <mem>256M</mem>
  <if id="1" net="Net1">
    <ipv4>10.0.1.1/24</ipv4>
    <ipv6>2001:0720:41c:4001::1/64</ipv6>
  </if>
  <if id="2" net="Net2">
    <ipv4>10.0.2.1/24</ipv4>
    <ipv6>2001:0720:41c:4002::1/64</ipv6>
  </if>
  <if id="3" net="Net3">
    <ipv4>10.0.3.1/24</ipv4>
    <ipv6>2001:0720:41c:4003::1/64</ipv6>
  </if>
  <if id="4" net="Net4">
    <ipv4>10.0.4.1/24</ipv4>
    <ipv6>2001:0720:41c:4004::1/64</ipv6>
  </if>
  <if id="5" net="Net0">
    <ipv4 mask="255.255.255.0">192.168.1.121</ipv4>
    </if>
    <route type="ipv4" gw="192.168.1.1">default</route>
    <forwarding type="ip" />
    <exec seq="dns" type="verbatim" mode="system">echo "nameserver 87.216.1.65" >
/etc/resolv.conf</exec>
    <exec seq="salida" type="verbatim" mode="system">iptables -t nat -A POSTROUTING -o eth5
-j MASQUERADE</exec>
    <exec seq="ruta1" type="verbatim" mode="system">route -A inet6 add 2001:720:41c:4005::/64
gw 2001:720:41c:4004::2 dev eth4</exec>
    <exec seq="ruta2" type="verbatim" mode="system">route -A inet6 add 64:FF9B::/96 gw
2001:720:41c:4004::2 dev eth4</exec>
  </vm>

<host>
  <hostif net="Net0">
    <ipv4 mask="255.255.255.0">192.168.1.125</ipv4>
  </hostif>

```

```
<physicalif name="eth0" type="ipv4" ip="192.168.1.125" mask="255.255.255.0"
gw="192.168.1.1"/>
  <route type="ipv4" gw="192.168.1.1">default</route>
</host>

</vnx>
```

Los servicios que se prueban en este escenario, tanto para IPv4 como para IPv6, son los siguientes:

- Configuración del protocolo.
- Encaminamiento.
- Configuración Stateful.
- Firewall.
- Servicio DNS.
- Servicio FTP.
- Servicio WEB.
- Servicio NAT64.

En la presente memoria se describen la instalación, configuración y pruebas realizadas para comprobar los siguientes servicios:

- Configuración del protocolo.
- Encaminamiento.
- Configuración Stateful.
- Firewall.

Mientras que los servicios DNS, WEB, FTP y NAT64 se describen en la memoria realizada por Gloria Martín Martín titulada "Diseño de escenarios de transición a IPv6 utilizando la herramienta VNX: DNS, servicios WEB y mecanismos de transición"

5.3 Servicios

5.3.1 Configuración del protocolo

Introducción

La mayor parte de los sistemas operativos tienen soporte para IPv4 e IPv6, aunque no siempre por defecto. Esto supone que en algunos casos es necesario instalar el protocolo.

En este apartado se hace una descripción de la instalación y configuración básica de IPv4 e IPv6 para los sistemas operativos:

- Windows XP SP1
- Ubuntu
- Fedora

Instalación

Veamos cómo se habilita el protocolo IPv6 en los diferentes sistemas operativos:

- **Windows XP SP3**

El protocolo IPv4 viene habilitado por defecto en el sistema operativo Windows XP SP3, no así el protocolo IPv6 que es necesario habilitarlo. Para ello existen dos procedimientos para habilitar IPv6:

1. Desde línea de comandos:

```
# ipv6 install
```

2. Desde la interfaz gráfica:

A través del panel de control, seleccione **“Conexiones de red”** y dentro de éste, seleccione **“Conexión de área local”**. El sistema le mostrará una pantalla en la que deberá seleccionar **“Propiedades”**. En la ventana que se muestra pulse **“Instalar”**, seleccione **“Protocolo”** y pulse **“Agregar”**. De los distintos protocolos que se pueden instalar, seleccione **“Microsoft TCP/IP versión 6”** y pulse **“Aceptar”**.

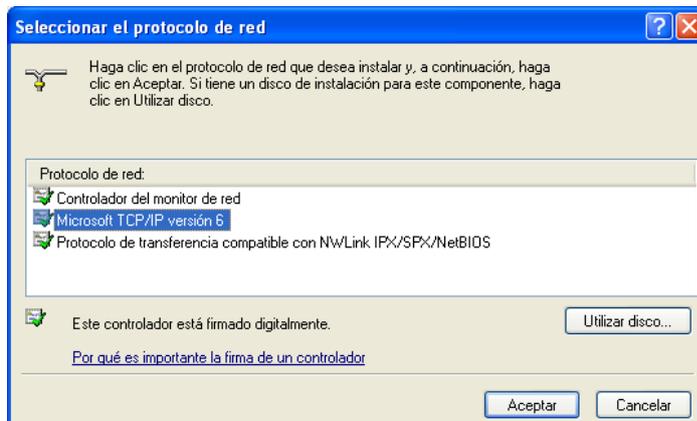


Figura 18. Instalación del protocolo IPv6 en Windows XP

- **Sistemas Linux**

En los sistemas Linux con versión de kernel superior a la 2.4.x están soportadas las versiones IPv4 e IPv6.

Configuración

Una vez instalados los protocolos, hay que configurarlos. Existen diferentes formas de configurar el protocolo IPv4 e IPv6 en un sistema. Las más habituales son:

1. **Desde línea de comando**

Esta opción es la más flexible ya que permite configurar las direcciones IPv4 e IPv6 de los sistemas ejecutando comandos previamente configurados en el fichero de configuración del entorno VNX.

2. **De forma automática mediante un servidor DHCP**

Esta opción se explicará más adelante en el servicio DHCP.

3. **De forma manual o a través de los ficheros de configuración.**

Veamos cómo se configura el protocolo IPv4 e IPv6 desde línea de comando y de la forma manual para los diferentes sistemas operativos utilizados en la simulación:

- **Windows XP SP3**

1. **Desde línea de comandos.**

Por ejemplo, para configurar el protocolo IPv4 en el interfaz “**Conexión de área local**” de “**Xp-2**”, ejecute la siguiente instrucción:

```
# netsh interface ip set address "Conexión de área local" static 10.0.2.2  
255.255.255.0 10.0.2.1 1
```

Para configurar el protocolo IPv6 del interfaz 2 de “**Xp-2**”, ejecute la siguiente instrucción:

```
# IPv6 adu 5/2001:720:41c:4002::2
```

2. **De forma manual.**

Con este método, sólo podemos configurar el protocolo IPv4. Para ello, a través del panel de control, seleccione “**Conexiones de red**” y dentro de éste, seleccione “**Conexión de área local**” y dentro de éste, seleccione “**Propiedades**”. En la ventana que se muestra seleccione

“Protocolo Internet (TCP/IP)”, seleccione “Usar la siguiente dirección IP”, tal y como muestra la siguiente figura:

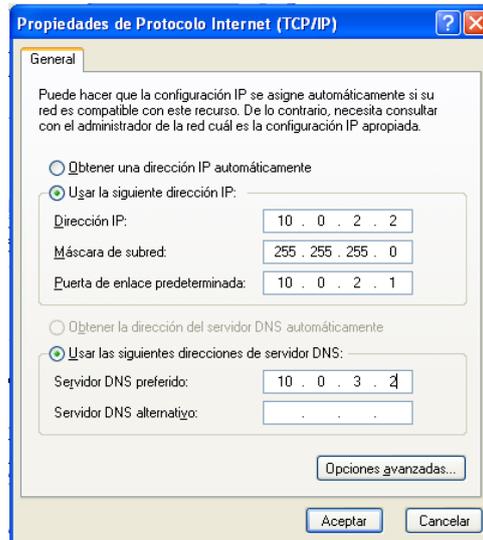


Figura 19. Configuración del protocolo IPv4 en Windows XP

Una vez finalizado, pulse “Aceptar”.

En sistemas operativos posteriores a Windows XP SP3 se puede configurar la dirección IPv6 de forma muy similar a como hemos configurado la dirección del protocolo IPv4.

- **Sistemas Linux**

1. **Desde comandos.**

Esta opción es la más flexible ya que permite configurar las máquinas ejecutando comandos previamente configurados en el fichero de configuración del entorno VNX. Para configurar una tarjeta con una dirección IPv6 e IPv4 habría que ejecutar los siguientes comandos:

- **Sistemas Ubuntu**

Por ejemplo, para configurar la dirección IPv4 en el servidor “Srv-web” ejecute el siguiente comando:

```
# ifconfig eth1 10.0.3.2 netmask 255.255.255.0
```

Para configurar la dirección IPv6 en el servidor “Srv-web” ejecute el siguiente comando:

```
# ip addr add 2001:720:41c:4003::2 dev eth1
```

- **Sistemas Fedora**

Por ejemplo, para configurar la dirección IPv4 en el servidor “**Srv-nat64**”:

```
# ifconfig eth1 10.0.4.2 255.255.255.0 up
```

Para configurar la dirección IPv6 en el servidor “**Srv-nat64**”:

```
# ifconfig eth1 add 2001:720:41c:4003::2
```

2. A través de los ficheros de configuración.

- **Sistemas Ubuntu**

```
# vi /etc/network/interfaces
```

```
# Interfaz lo
```

```
auto lo
```

```
iface lo inet loopback
```

```
# Interfaz eth1
```

```
auto eth1
```

```
iface eth1 inet static
```

```
address 10.0.3.2
```

```
netmask 255.255.255.0
```

```
iface eth1 inet6 static
```

```
address 2001:0720:41c:4003::2
```

```
netmask 64
```

- **Sistemas Fedora**

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth1
```

```
HWADDR=02:fd:00:00:08:01
```

```
TYPE=Ethernet
```

```
ONBOOT=yes
```

```
NAME="eth1"
```

```
IPV6INIT=yes
```

```
IPADDR=10.0.4.2
```

```
NETMASK=255.255.255.0
```

```
IPV6_AUTOCONF=no
```

Si no hemos configurado la ruta por defecto en los ficheros de configuración, podemos ejecutar la siguiente instrucción:

- **Sistemas Windows SP3**

Para establecer la ruta por defecto en IPv4 en “Xp-2” ejecutamos el siguiente comando:

```
# route add 0.0.0.0 mask 0.0.0.0 10.0.2.1 metric 1
```

Para establecer la ruta por defecto en IPv6 en “Xp-2” ejecutamos el siguiente comando:

```
# netsh interface IPv6 add route ::/0 5 2001:720:41c:4002::1
```

- **Sistemas Ubuntu**

Para establecer la ruta por defecto en IPv4 en “Srv-web” ejecutamos el siguiente comando:

```
# route add default gw 10.0.3.1 eth1
```

Para establecer la ruta por defecto en IPv6 en “Srv-web” ejecutamos el siguiente comando:

```
# route -A inet6 add default gw 2001:720:41c:4003::1
```

- **Sistemas Fedora**

Para establecer la ruta por defecto en IPv4 en “Srv-nat64” ejecutamos el siguiente comando:

```
# route add -net 0.0.0.0 netmask 0.0.0.0 gw 10.0.4.1
```

Para establecer la ruta por defecto en IPv6 en “Srv-nat64” ejecutamos el siguiente comando:

```
# route -A inet6 add ::/0 gw 2001:720:41c:4004::1
```

Pruebas

Para comprobar la configuración del protocolo IPv4 e IPv6 en los diferentes equipos del escenario, basta con mostrar la configuración de los interfaces. Adicionalmente se puede comprobar si se llega a la red haciendo un “ping” al “Gateway”.

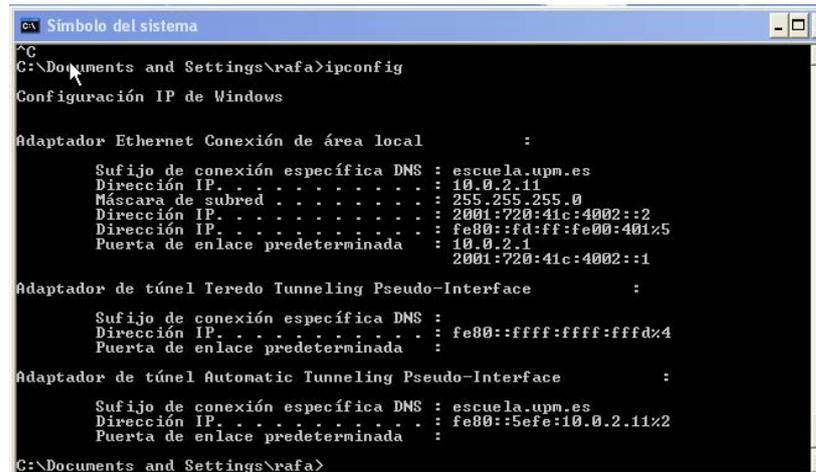
La forma de hacerlo en función del sistema operativo es la siguiente:

- Windows XP SP3

Para ver la configuración IPv4 de las tarjetas de red sólo tiene que ejecutar la siguiente instrucción desde línea de comandos:

```
# ipconfig
```

El resultado debería ser como el que se muestra en la siguiente figura:



```
^C
C:\Documents and Settings\rafa>ipconfig

Configuración IP de Windows

Adaptador Ethernet Conexión de área local      :
    Sufijo de conexión específica DNS : escuela.upm.es
    Dirección IP. . . . . : 10.0.2.11
    Máscara de subred . . . . . : 255.255.255.0
    Dirección IP. . . . . : 2001:720:41c:4002::2
    Dirección IP. . . . . : fe80::fd:ff:fe00:401%5
    Puerta de enlace predeterminada : 10.0.2.1
                                     2001:720:41c:4002::1

Adaptador de túnel Teredo Tunneling Pseudo-Interface      :
    Sufijo de conexión específica DNS :
    Dirección IP. . . . . : fe80::ffff:ffff:ffff:4
    Puerta de enlace predeterminada :

Adaptador de túnel Automatic Tunneling Pseudo-Interface      :
    Sufijo de conexión específica DNS : escuela.upm.es
    Dirección IP. . . . . : fe80::5efe:10.0.2.11%2
    Puerta de enlace predeterminada :

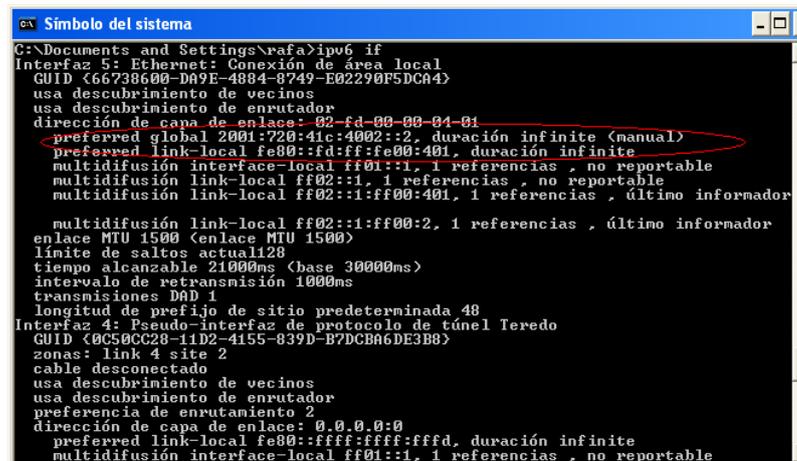
C:\Documents and Settings\rafa>
```

Figura 20. Comprobación de la configuración IPv4 en "Xp-2"

Para ver la configuración IPv6 de las tarjetas de red sólo tenemos que ejecutar la siguiente instrucción desde línea de comandos:

```
# ipv6 if
```

El resultado debería ser como el que se muestra en la siguiente figura:



```
^C
C:\Documents and Settings\rafa>ipv6 if

Interfaz 5: Ethernet: Conexión de área local
GUID {66738600-DA9E-4884-8749-E02290F5DCA4}
usa descubrimiento de vecinos
usa descubrimiento de enrutador
dirección de capa de enlace: fd-00-00-04-01
preferred global 2001:720:41c:4002::2, duración infinite (manual)
preferred link-local fe80::fd:ff:fe00:401, duración infinite
multidifusión interface-local ff01::1, 1 referencias , no reportable
multidifusión link-local ff02::1, 1 referencias , no reportable
multidifusión link-local ff02::1:ff00:401, 1 referencias , último informador

multidifusión link-local ff02::1:ff00:2, 1 referencias , último informador
enlace MTU 1500 (enlace MTU 1500)
límite de saltos actual128
tiempo alcanzable 21000ms (base 30000ms)
intervalo de retransmisión 1000ms
transmisiones DAD 1
longitud de prefijo de sitio predeterminada 48

Interfaz 4: Pseudo-interfaz de protocolo de túnel Teredo
GUID {0C50CC28-11D2-4155-839D-B7DCBA6DE3B8}
zonas: link 4 site 2
cable desconectado
usa descubrimiento de vecinos
usa descubrimiento de enrutador
preferencia de enrutamiento 2
dirección de capa de enlace: 0.0.0.0
preferred link-local fe80::ffff:ffff:ffff:4, duración infinite
multidifusión interface-local ff01::1, 1 referencias , no reportable

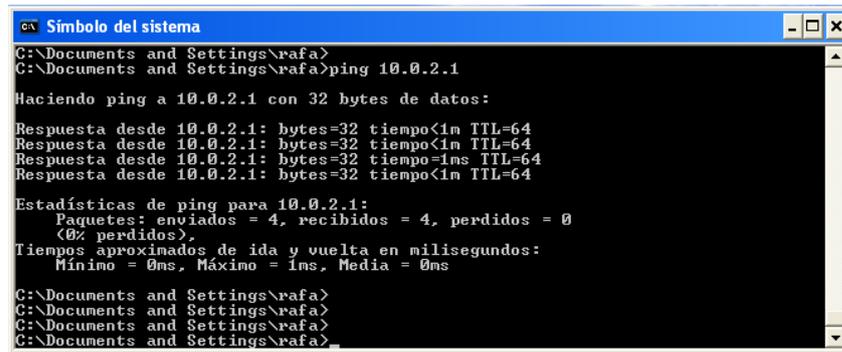
C:\Documents and Settings\rafa>
```

Figura 21. Comprobación de la configuración IPv6 en "Xp-2"

Para comprobar la conectividad con IPv4 hacemos un “ping” al “Gateway”. Por ejemplo, para comprobar el acceso a la red desde “Xp-2” ejecutamos el siguiente comando:

```
# ping 10.0.2.1
```

El resultado debería ser como el que se muestra en la siguiente figura:



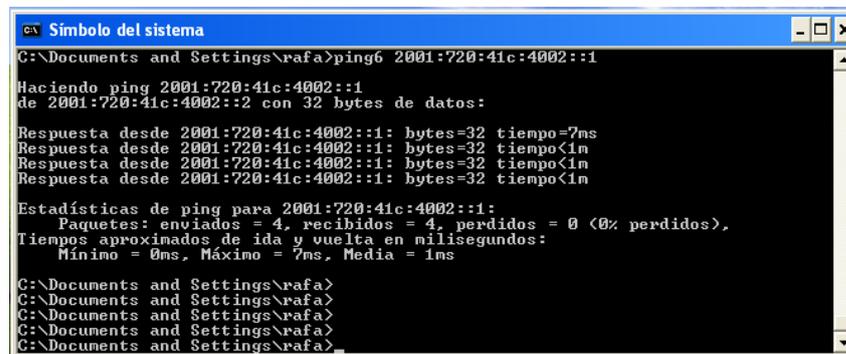
```
Simbolo del sistema
C:\Documents and Settings\rafa>
C:\Documents and Settings\rafa>ping 10.0.2.1
Haciendo ping a 10.0.2.1 con 32 bytes de datos:
Respuesta desde 10.0.2.1: bytes=32 tiempo<1m TTL=64
Respuesta desde 10.0.2.1: bytes=32 tiempo<1m TTL=64
Respuesta desde 10.0.2.1: bytes=32 tiempo=1ms TTL=64
Respuesta desde 10.0.2.1: bytes=32 tiempo<1m TTL=64
Estadísticas de ping para 10.0.2.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (<0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 1ms, Media = 0ms
C:\Documents and Settings\rafa>
C:\Documents and Settings\rafa>
C:\Documents and Settings\rafa>
C:\Documents and Settings\rafa>
```

Figura 22. Comprobación de la conectividad IPv4 en “Xp-2”

Para comprobar la conectividad con IPv6 hacemos un “ping” al “Gateway”. Por ejemplo, para comprobar el acceso a la red desde “Xp-2” ejecutamos el siguiente comando:

```
# ping6 2001:720:41c:4002::1
```

El resultado debería ser como el que se muestra en la siguiente figura:



```
Simbolo del sistema
C:\Documents and Settings\rafa>ping6 2001:720:41c:4002::1
Haciendo ping 2001:720:41c:4002::1
de 2001:720:41c:4002::2 con 32 bytes de datos:
Respuesta desde 2001:720:41c:4002::1: bytes=32 tiempo=7ms
Respuesta desde 2001:720:41c:4002::1: bytes=32 tiempo<1m
Respuesta desde 2001:720:41c:4002::1: bytes=32 tiempo<1m
Respuesta desde 2001:720:41c:4002::1: bytes=32 tiempo<1m
Estadísticas de ping para 2001:720:41c:4002::1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (<0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 7ms, Media = 1ms
C:\Documents and Settings\rafa>
C:\Documents and Settings\rafa>
C:\Documents and Settings\rafa>
C:\Documents and Settings\rafa>
```

Figura 23. Comprobación de la conectividad IPv6 en “Xp-2”

- **Fedora y Ubuntu**

La forma de ver la configuración de la red en un sistema Linux es igual, independientemente de que la distribución sea Ubuntu o Fedora.

Para ver la configuración de las tarjetas de red, por ejemplo para “Srv-nat64”, sólo tenemos que ejecutar la siguiente instrucción desde línea de comandos:

```
# ifconfig -a

eth1  Link encap: Ethernet HWaddr 02:FD:00:00:08:01
inet addr:10.0.4.2 Bcast:10.0.4.255 Mask:255.255.255.0
inet6addr: fe80::fd:ff:fe00:801/64 Scope:Link
inet6 addr: 2001:720:41c:4004::2/64 Scope:Global
    UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
    RX packets:5407 errors:0 dropped:0 overruns:0 frame:0
    TX packets:5061 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
    RX bytes:1460488 (1.3 MiB)  TX bytes:889263 (868.4 KiB)
Interrupt:11 Base address:0x6000

eth2  Link encap: Ethernet HWaddr 02:FD:00:00:08:02
inet6addr: fe80::fd:ff:fe00:802/64 Scope:Link
inet6 addr: 2001:720:41c:4005::1/64 Scope:Global
    UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
    RX packets:1490 errors:0 dropped:0 overruns:0 frame:0
    TX packets:1419 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
    RX bytes:177176 (173.0 KiB)  TX bytes:169176 (165.2 KiB)
Interrupt:11 Base address:0xa000

lo    Link encap: Local Loopback
inet addr:127.0.0.1  Mask:255.0.0.0
inet6addr: ::1/128 Scope:Host
    UP LOOPBACK RUNNING  MTU:16436  Metric:1
    RX packets:139 errors:0 dropped:0 overruns:0 frame:0
    TX packets:139 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
    RX bytes:18478 (18.0 KiB)  TX bytes:18478 (18.0 KiB)

nat64  Link encap: UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
00
    UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
    RX packets:523 errors:0 dropped:0 overruns:0 frame:0
    TX packets:549 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
    RX bytes:68860 (67.2 KiB)  TX bytes:416311 (406.5 KiB)
```

Para comprobar que se puede acceder a la red hacemos un “ping” al “Gateway”. Por ejemplo, para comprobar el acceso a la red desde “Srv-nat64” ejecutamos el siguiente comando:

```
# ping 10.0.4.1
PING 10.0.4.1 (10.0.4.1) 56(84) bytes of data.
64 bytes from 10.0.4.1: icmp_req=1 ttl=64 time=7.33 ms
64 bytes from 10.0.4.1: icmp_req=2 ttl=64 time=0.626 ms
64 bytes from 10.0.4.1: icmp_req=3 ttl=64 time=0.501 ms
64 bytes from 10.0.4.1: icmp_req=4 ttl=64 time=0.477 ms
64 bytes from 10.0.4.1: icmp_req=5 ttl=64 time=0.470 ms
--- 10.0.4.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4001ms
rtt min/avg/max/mdev = 0.470/1.881/7.333/2.726 ms

# ping6 2001:720:41c:4004::1
PING 2001:720:41c:4004::1(2001:720:41c:4004::1) 56 data bytes
64 bytes from 2001:720:41c:4004::1: icmp_seq=1 ttl=64 time=0.696 ms
64 bytes from 2001:720:41c:4004::1: icmp_seq=2 ttl=64 time=0.523 ms
64 bytes from 2001:720:41c:4004::1: icmp_seq=3 ttl=64 time=0.558 ms
64 bytes from 2001:720:41c:4004::1: icmp_seq=4 ttl=64 time=0.457 ms
--- 2001:720:41c:4004::1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 6000ms
rtt min/avg/max/mdev = 0.457/0.568/0.696/0.085 ms
```

5.3.2 Encaminamiento

Introducción

La asignación de direcciones en IPv4 e IPv6 se hace de forma jerárquica, luego el encaminamiento tanto en IPv4 como en IPv6 es jerárquico.

Para encaminar los paquetes se requiere previamente el establecimiento de rutas en los routers. Para llevar a cabo esta tarea se ha optado por utilizar rutas estáticas debido a que la topología de la red varía poco. En este caso, será el administrador de la red quién establece las rutas que definen el comportamiento de los routers.

Instalación

De acuerdo con la topología de la red descrita en el escenario de simulación, los equipos que se tienen que enrutar paquetes son: “Gateway” (enruta IPv4 e IPv6) y “Srv-nat64” (sólo enruta IPv6).

Los sistemas operativos Linux pueden comportarse como un router, sólo es necesario configurarlos para ello.

Configuración

Para configurar un sistema Linux para que se comporte como un router hay que modificar el fichero `/etc/sysctl.conf` de la siguiente forma:

```
# Controls IP packet forwarding
net.ipv4.ip_forward = 1
net.IPv6.conf.all.forwarding=1
```

Además, si es necesario hay que establecer rutas para que haya conectividad entre todas las máquinas. De acuerdo a la topología de la red del escenario vemos que es necesario establecer una ruta IPv6 en "Gateway". Para configurar la ruta se ejecuta el siguiente comando:

```
# route -A inet6 add 2001:720:41c:4005::/64 gw 2001:720:41c:4004::2 dev eth4
```

Pruebas

Para comprobar que un sistema enruta los paquetes de una red a otra podemos establecer una conexión entre dos máquinas conectadas a distintas redes, o bien, hacer una traza de los paquetes.

La forma de hacerlo en función del sistema operativo es la siguiente:

- **Windows XP SP3**

Para ver si los paquetes IPv4 llegan desde la máquina "Xp-2" hasta "Srv-nat64" sólo tenemos que ejecutar la siguiente instrucción desde línea de comandos:

```
# tracert 10.0.4.2
```

El resultado debería ser como el que se muestra en la siguiente figura:

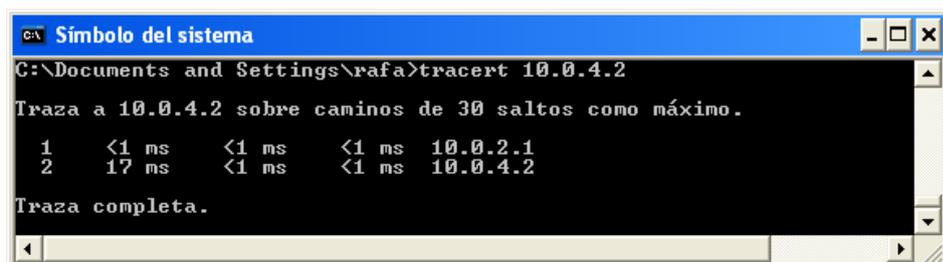


Figura 24. Comprobación del enrutamiento en IPv4 en "Xp-2"

Para ver si los paquetes IPv6 llegan desde la máquina “Xp-2” hasta “Srv-nat64” sólo tenemos que ejecutar la siguiente instrucción desde línea de comandos:

```
# tracert6 2001:720:41c:4002::2
```

El resultado debería ser como el que se muestra en la siguiente figura:

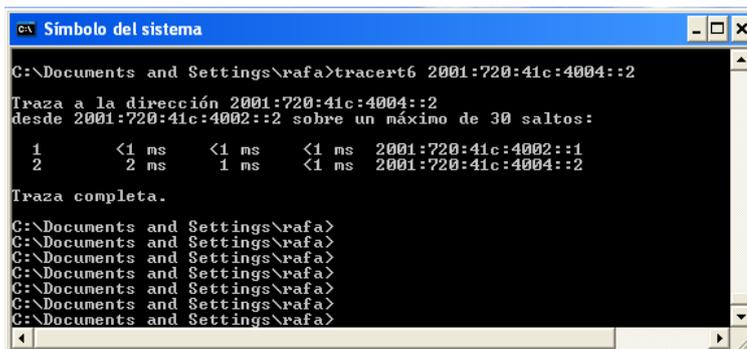


Figura 25. Comprobación del enrutamiento en IPv6 en “Xp-2”

- **Fedora y Ubuntu**

Para ver si los paquetes llegan desde la máquina “Srv-web” hasta “Srv-nat64” sólo tenemos que ejecutar, desde “Srv-web” la siguiente instrucción desde línea de comandos:

```
# traceroute 10.0.4.2
traceroute to 10.0.4.2 (10.0.4.2), 30 hops max, 60 byte packets
 1 10.0.3.1 (10.0.3.1) 0.597 ms 0.367 ms 0.301 ms
 2 10.0.4.2 (10.0.4.2) 23.205 ms 22.794 ms 22.544 ms
#tracert6 2001:720:41c:4004::2
traceroute to 2001:720:41c:4004::2 (2001:720:41c:4004::2) from 2001:720:41c:4003::2, 30
hops max, 16 byte packets
 1 2001:720:41c:4003::1 (2001:720:41c:4003::1) 32.54 ms 0.377 ms 0.293 ms
 2 2001:720:41c:4004::2 (2001:720:41c:4004::2) 0.717 ms 0.638 ms 0.643 ms
#
```

5.3.3 Configuración Stateful

Introducción

Para configurar los parámetros de los protocolos IPv4 e IPv6 de los clientes de forma automática hemos optado por utilizar un servidor DHCP (Dynamic Host Config Protocol). DHCP es un protocolo UDP cliente/servidor, diseñado para reducir el coste

de gestión de nodos en entornos donde los administradores precisan un control sobre la asignación de los recursos de la red.

Para lograr este objetivo, se centraliza la gestión de los recursos de la red tales como direcciones IP, información de encaminamiento, información de instalación de Sistemas Operativos, información de servicios de directorios, etc., sobre uno o varios servidores DHCP, en lugar de distribuir dicha información en ficheros de configuración locales en cada nodo.

Además, DHCP ha sido diseñado para ser fácilmente extensible con nuevos parámetros de configuración, a través de “extensiones” que incorporan esta nueva información.

La autoconfiguración “stateful” en IPv6 permite al host obtener parámetros de forma automática que no podría obtener mediante mecanismos de configuración “stateless”, tales como el servidor DNS, el dominio al que pertenece, los servidores de hora, etc. No obstante, ambos mecanismos son compatibles y pueden usarse de forma concurrente para reducir el coste de propiedad y administración de la red.

En nuestro caso vamos a utilizar un mecanismo RADVD (Router Advertisement Daemon) para configurar la ruta por defecto a los clientes. RADVD es un demonio que actúa en routers IPv6, los cuales envían mensajes RA (Router Advertisement, [RFC 2461]) a una red, de forma periódica. Cuando llega a un nodo, este envía mensajes de RS (Router Solicitation). Estos mensajes se utilizan en la configuración IPv6 stateless.

Instalación

De acuerdo con la topología de la red descrita en el escenario de simulación, el equipo que se comporta como un servidor DHCP es “**Srv-dhcp**” y hay varios clientes que reciben la configuración de los protocolos por DHCP como “**Xp-1**”, “**Ciente-Ubuntu**” y “**Xp-2**”. Como los clientes están conectados en una red diferente a la que está el servidor DHCP, es necesario que un equipo se comporte como relay de DHCP, este equipo es “**Gateway**”.

Además de comportarse como relay de dhcp, “**Gateway**” debe escuchar las solicitudes IPv6 de enrutadores procedentes de los clientes, ya que el servidor DHCP de IPv6, Dibbler [15], no ofrece como parámetro la ruta por defecto. Por este motivo tenemos que instalar RADVD en “**Gateway**”. Para instalar RADVD sólo tenemos que ejecutar el siguiente comando:

```
# apt-get install radvd
```

Una vez instalado, sólo faltaría configurar el servicio, tal y como veremos en el siguiente apartado.

Como software para servidor DHCP hemos instalado `dhcp3-server` para IPv4 y `Dibbler` para IPv6. Veamos los pasos que hay que seguir para instalar el servidor, el cliente y el relay de DHCP.

- **Instalación del Servidor DHCP**

El sistema operativo instalado en el servidor DHCP es Ubuntu por lo que para instalar el servidor debemos proceder de la siguiente forma:

- **Instalación de `dhcp3-server`**

Para instalar en “`Srv-dhcp`” el paquete `dhcp3-server` sólo tenemos que ejecutar el siguiente comando:

```
# apt-get install dhcp3-server
```

- **Instalación de `dibbler-server`**

Para instalar en “`Srv-dhcp`” el paquete `dibbler-server` sólo tenemos que ejecutar el siguiente comando:

```
# apt-get install dibbler-server
```

- **Instalación del Relay DHCP**

El sistema operativo instalado en el servidor DHCP es Ubuntu por lo que para instalar el servidor debemos proceder de la siguiente forma:

- **Instalación de `dhcp3-relay`**

Para instalar en “`Gateway`” el paquete `dhcp3-relay` sólo tenemos que ejecutar el siguiente comando:

```
# apt-get install dhcp3-relay
```

- **Instalación de `dibbler-relay`**

Para instalar en “`Gateway`” el paquete `dibbler-relay` sólo tenemos que ejecutar el siguiente comando:

```
# apt-get install dibbler-relay
```

- **Instalación de clientes DHCP**
 - **Instalación del cliente dhcp**

En este caso no hay que instalar nada en los clientes. Ya viene instalado por defecto, tanto en sistemas Windows como en sistemas Linux.

- **Instalación de dibbler-client**
 - **Windows XP SP3**

La instalación en Windows resulta sencilla. Sólo debe descargarse el paquete “**dibbler-0.7.2-win32.exe**”.

Una vez descargado el programa, debe ejecutar el programa de instalación y seguir las instrucciones que nos brinda el asistente, seleccionando que queremos instalar al DHCPv6 client, tal como se muestra en la siguiente figura:

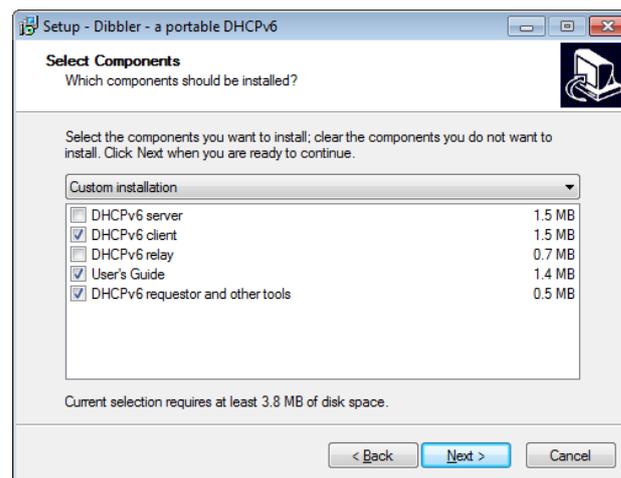


Figura 26. Selección de paquetes a instalar en el asistente de Dibbler”

Una vez instalado, aparecerá un menú en el menú inicio a través del cual podremos realizar todas las funciones de inicio y parada del servicio Dibbler. Si lo deseamos, podemos instalar el Dibbler-Client como un servicio para que inicie automáticamente con el sistema.

Una vez instalados, la configuración necesaria para lograr las funcionalidades básicas es mínima.

- **Ubuntu**

Para instalar en un cliente Ubuntu el paquete `dibbler-client` sólo tenemos que ejecutar el siguiente comando:

```
# apt-get install dibbler-client
```

Configuración

- **RADVD**

Para configurar el demonio RADVD debe crear un fichero llamado `/etc/radvd.conf` con el siguiente contenido:

```
interface eth1
{
  AdvSendAdvert on;
  MinRtrAdvInterval 3;
  MaxRtrAdvInterval 10;
  prefix 2001:720:41c:4001::/64
  {
    AdvOnLink on;
    AdvAutonomous on;
    AdvRouterAddr off;
  };
};

interface eth2
{
  AdvSendAdvert on;
  MinRtrAdvInterval 3;
  MaxRtrAdvInterval 10;
  prefix 2001:720:41c:4002::/64
  {
    AdvOnLink on;
    AdvAutonomous on;
    AdvRouterAddr off;
  };
};
```

- DHCP
 - Cliente
 - Dhcp3
 - **Sistemas Windows Xp**

Para configurar que un sistema Windows XP configure los parámetros del protocolo IPv4 a través de un servidor DHCP debemos seguir los siguientes pasos: a través del panel de control, seleccione “**Conexiones de red**” y dentro de éste, seleccione “**Conexión de área local**” y dentro de éste, seleccione “**Propiedades**”. En la ventana que se muestra seleccione “**Protocolo Internet (TCP/IP)**”, seleccione “**Usar la siguiente dirección IP**”, tal y como muestra la siguiente figura:

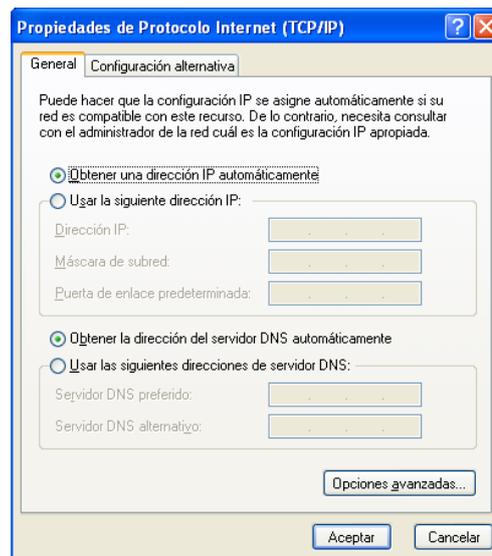


Figura 27. Configuración del protocolo IPv4 por DHCP en Windows XP

Una vez finalizado, pulse “**Aceptar**”.

- **Sistemas Ubuntu**

Para configurar que un cliente configure los parámetros del protocolo IPv4 a través de un servidor DHCP debemos editar el fichero /etc/network/interfaces y poner lo siguiente:

```
# vi /etc/network/interfaces
auto eth1
```

```
iface eth1 inet dhcp
```

Una vez configurado, debemos reiniciar el servicio de red:

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

○ **Dibbler-client**

▪ **Sistemas Windows Xp**

Para configurar que un cliente dibbler en un sistema Windows XP debemos editar el fichero `c:/dibbler/client.conf` y poner lo siguiente:

```
# Defaults for dibbler-client.  
Log-mode short  
log-level 7  
iface "Conexión de área local" {  
# ask for options  
option dns-server  
option domain  
# ask for address  
ia  
}
```

Una vez configurado, debemos iniciar el servicio y podemos hacerlo en modo servicio o en modo consola.

▪ **Sistemas Ubuntu**

Editamos el fichero `/etc/dibbler/client.conf` y hacemos los siguientes cambios:

```
# Defaults for dibbler-client.  
log-level 7  
iface eth0 {  
# ask for address  
Ia  
# ask for options  
option dns-server  
option domain  
}
```

Una vez configurado, debemos iniciar el servicio:

```
# /etc/init.d/dibbler-client start
```

- **Relay**

- **Dhcp3-relay**

Editamos el fichero `/etc/default/dhcp3-relay` y hacemos los siguientes cambios:

```
# Servidor al que se le envían las peticiones
SERVERS="10.0.3.3"
# interfaces por las que escucha peticiones
INTERFACES="eth1 eth2 eth3"
# Additional options that are passed to the DHCP relay daemon?
OPTIONS=""
```

Una vez configurado, debe iniciar el servicio:

```
# /etc/init.d/dhc3-relay start
```

- **Dibbler-relay**

Editamos el fichero `/etc/dibbler/relay.conf` y hacemos los siguientes cambios:

```
log-level 8
log-mode short
iface eth3 {
server multicast yes
}
iface eth1 {
client multicast yes
interface-id 5020
}
iface eth2 {
client multicast yes
interface-id 5021
}
```

Una vez configurado, debemos iniciar el servicio:

```
# /etc/init.d/dibbler-relay start
```

- **Servidor**
 - **Dhcp3-server**

Editamos el fichero de configuración /etc/dhcp3/dhcpd.conf y hacemos los siguientes cambios:

```
#####  
# Configuración de los parámetros generales  
#####  
# Servidor autorizado  
authoritative;  
# Define el modo de esquema DNS de actualización  
ddns-update-style none;  
# Define el dominio DNS  
option domain-name "escuela.upm.es";  
# Define los servidores de DNS  
Option domain-name-servers 10.0.3.4;  
# Duración por defecto de la concesión en segundos  
default-lease-time 600;  
# Duración máxima de la concesión en segundos  
max-lease-time 7200;  
# Logs del servicio DHCP.  
log-facility local7;  
#####  
# NET2  
#####  
subnet 10.0.2.0 netmask 255.255.255.0 {  
option broadcast-address 10.0.2.255;  
option routers 10.0.2.1;  
range 10.0.2.2 10.0.2.254;  
}  
#####  
# NET1  
#####  
subnet 10.0.1.0 netmask 255.255.255.0 {  
option broadcast-address 10.0.1.255;  
option routers 10.0.1.1;
```

```
range 10.0.1.2 10.0.1.254;
}
```

Una vez configurado, debemos iniciar el servicio:

```
# /etc/init.d/dhcp3-server start
```

o Dnsmasq

Editamos el fichero `/etc/dnsmasq/dnsmasq.conf` y hacemos los siguientes cambios:

```
# Logging level range: 1(Emergency)-8(Debug)
log-level 8
# Don't log full date
log-mode short
# Indica que este servidor tiene mayor preferencia que cualquier
# otro servidor con menor valor en este parámetro.
preference 0
# stateless
guess-mode
# T1 define el tiempo que deben esperar los clientes para comenzar
# el proceso de renovación de direcciones.
t1 1800-2000
# T2 define el tiempo que deben esperar los clientes para el
# reintento después de un fallo en el proceso de renovación.
t2 2700-3000
# Define el tiempo en el que la dirección asignada es usada como
# "preferred".
preferred-lifetime 3600
# Define el tiempo en el que la dirección asignada es válida.
valid-lifetime 7200
iface relay1 {
  relay eth1
  interface-id 5020
  class {
    pool 2001:720:041c:4001::/64
  }
# Define los servidores de DNS
option dns-server 2001:720:041c:4003::4
# Define el dominio DNS
option domain escuela.upm.es
```

```

# provide vendor-specific data (vendor-id set to 5678)
option vendor-spec 5678-0x0002aaaa
#Define los servidores de hora
option ntp-server 2000::200,2000::201,2000::202
# Información de zona horaria
option time-zone CET
}

iface relay1 {
  relay eth1
  interface-id 5021
  class {
    pool 2001:720:041c:4002::/64
  }
# Define los servidores de DNS
option dns-server 2001:720:041c:4003::4
# Define el dominio DNS
option domain escuela.upm.es
# provide vendor-specific data (vendor-id set to 5678)
option vendor-spec 5678-0x0002aaaa
#Define los servidores de hora
option ntp-server 2000::200,2000::201,2000::202
# Información de zona horaria
option time-zone CET
}

```

Una vez configurado, debe iniciar el servicio:

```
# /etc/init.d/dnsmasq start
```

Pruebas

Para probar el funcionamiento correcto tanto del servicio DHCP como del servicio RADVD, basta con comprobar que los protocolos IPv4 e IPv6 de los sistemas operativos clientes se configuran de forma apropiada.

- **Windows XP**

Para comprobar que un cliente Windows XP se ha configurado el protocolo IPv4 por DHCP, podemos ver la configuración del protocolo ejecutando el comando “**ipconfig**”. En el caso del equipo “**XP-1**” el sistema nos mostrará lo siguiente:

```

C:\Documents and Settings\rafa>
Adaptador Ethernet Conexión de área local :
    Sufijo de conexión específica DNS : escuela.upm.es
    Dirección IP. . . . . : 10.0.1.2
    Máscara de subred . . . . . : 255.255.255.0
    Dirección IP. . . . . : 2001:720:41c:4001:a112:d571:dfed:a6c
6
    Dirección IP. . . . . : fe80::fd:ff:fe00:101%5
    Puerta de enlace predeterminada : 10.0.1.1
Adaptador de túnel Teredo Tunneling Pseudo-Interface :
    Sufijo de conexión específica DNS :
    Dirección IP. . . . . : fe80::ffff:ffff:ffffd%4
    Puerta de enlace predeterminada :
Adaptador de túnel Automatic Tunneling Pseudo-Interface :
    Sufijo de conexión específica DNS : escuela.upm.es
    Dirección IP. . . . . : fe80::5efe:10.0.1.2%2
    Puerta de enlace predeterminada :
C:\Documents and Settings\rafa>

```

Figura 28. Configuración de los protocolos IPv4 e IPv6 en un interfaz de red en Windows XP

También podemos comprobar la conectividad del equipo haciendo un ping a una máquina conectada en la red, por ejemplo, a la dirección IPv4 10.0.4.2:

```

C:\Documents and Settings\rafa>
Dirección IP. . . . . : fe80::ffff:ffff:ffffd%4
Puerta de enlace predeterminada :
Adaptador de túnel Automatic Tunneling Pseudo-Interface :
    Sufijo de conexión específica DNS : escuela.upm.es
    Dirección IP. . . . . : fe80::5efe:10.0.1.2%2
    Puerta de enlace predeterminada :
C:\Documents and Settings\rafa>ping 10.0.4.2
Haciendo ping a 10.0.4.2 con 32 bytes de datos:
Respuesta desde 10.0.4.2: bytes=32 tiempo=3ms TTL=63
Respuesta desde 10.0.4.2: bytes=32 tiempo=1ms TTL=63
Respuesta desde 10.0.4.2: bytes=32 tiempo=1ms TTL=63
Respuesta desde 10.0.4.2: bytes=32 tiempo=1ms TTL=63
Estadísticas de ping para 10.0.4.2:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (<0% perdidos>),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 1ms, Máximo = 3ms, Media = 1ms
C:\Documents and Settings\rafa>

```

Figura 29. Comprobación de la conectividad a nivel IPv4 en un interfaz de red en Windows XP

La asignación de la dirección IPv4 por parte del servidor al cliente se puede comprobar viendo las asignaciones del servidor dhcp almacenadas en el fichero `/var/lib/dhcp3/dhcpd.leases`:

```

# The format of this file is documented in the dhcpd.leases (5) manual page.
# This lease file was written by isc-dhcp-V3.1.3
lease 10.0.1.2 {
starts 0 2011/07/03 12:05:04;
ends 0 2011/07/03 12:15:04;
cltt 0 2011/07/03 12:05:04;
binding state active;
next binding state free;
hardware ethernet 02:fd:00:00:01:01;
uid "\001\002\375\000\000\001\001";

```

```

client-hostname "Xp-1";
}
lease 10.0.1.3 {
starts 0 2011/07/03 12:06:27;
ends 0 2011/07/03 12:16:27;
cltt 0 2011/07/03 12:06:27;

binding state active;

next binding state free;

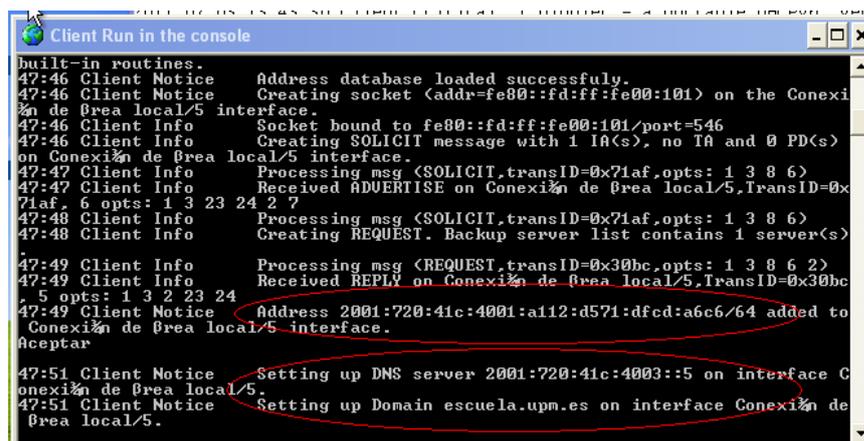
hardware ethernet 02:fd:00:00:02:01;

client-hostname "Cliente-Ubuntu";
}

```

Para comprobar que un cliente Windows XP se ha configurado el protocolo IPv6 por DHCP y RADVD, podemos comprobar que el sistema Windows se configura con una dirección IPv6 y que además puede acceder a otros sistemas que no se encuentren en la misma red que él.

Para comprobar que el sistema Windows se configura a través de un servidor DHCP, podemos hacer una petición de IPv6 al servidor DHCP mediante el cliente dibbler. En este caso, desde el equipo "Xp-1", debemos acceder a "Inicio", seleccionar "Todos los programas" y dentro de éste, seleccionamos "Dibbler". En las opciones que se muestran, seleccionamos "Client Run in the console". Para el caso del equipo "XP-1" la pantalla que mostrará el sistema es la siguiente:



```

built-in routines.
47:46 Client Notice Address database loaded successfully.
47:46 Client Notice Creating socket (addr=fe80::fd:ff:fe00:101) on the Conexi
ción de área local/5 interface.
47:46 Client Info Socket bound to fe80::fd:ff:fe00:101/port=546
47:46 Client Info Creating SOLICIT message with 1 IA(s), no IA and 0 PD(s)
on Conexión de área local/5 interface.
47:47 Client Info Processing msg (SOLICIT,transID=0x71af,opts: 1 3 8 6)
47:47 Client Info Received ADVERTISE on Conexión de área local/5,TransID=0x
71af, 6 opts: 1 3 23 24 2 7
47:48 Client Info Processing msg (SOLICIT,transID=0x71af,opts: 1 3 8 6)
47:48 Client Info Creating REQUEST. Backup server list contains 1 server(s)
47:49 Client Info Processing msg (REQUEST,transID=0x30bc,opts: 1 3 8 6 2)
47:49 Client Info Received REPLY on Conexión de área local/5,TransID=0x30bc
, 5 opts: 1 3 2 23 24
47:49 Client Notice Address 2001:720:41c:4001:a112:d571:dfcd:a6c6/64 added to
Conexión de área local/5 interface.
Aceptar
47:51 Client Notice Setting up DNS server 2001:720:41c:4003::5 on interface C
onexión de área local/5.
47:51 Client Notice Setting up Domain escuela.upm.es on interface Conexión de
área local/5.

```

Figura 30. Funcionamiento del sistema cliente de Dibbler en "Xp-1"

En la figura anterior se puede observar que el servidor le asigna una dirección IPv6 y parámetros como el servidor de DNS y el dominio. La dirección IPv6 se puede mostrar ejecutando el comando "ipconfig":

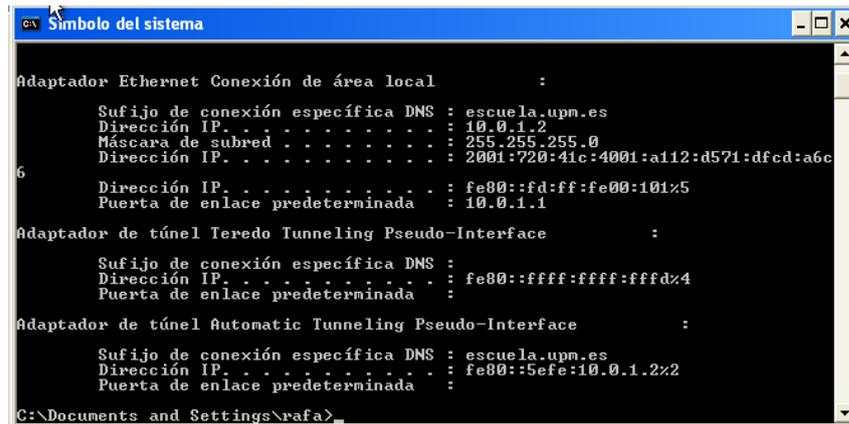


Figura 31. Configuración de IPv6 del sistema “Xp-1”

Para comprobar que RADVD le ha configurado el router por defecto correctamente, al igual que hacíamos para el protocolo IPv4, podemos hacer un ping. Si queremos comprobar la conectividad en IPv6 desde “Xp1” hasta “Srv-nat64”, el sistema nos mostrará una pantalla como la siguiente:

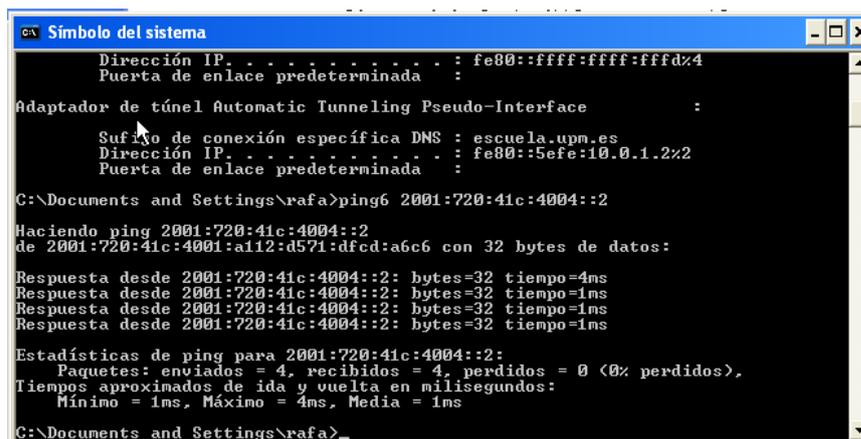


Figura 32. Comprobación de la conectividad en IPv6 desde “Xp-1” hasta “Srv-nat64”

- **Ubuntu**

Si queremos hacer las comprobaciones con un cliente con sistema Ubuntu, el procedimiento es muy similar.

Para comprobar que se ha configurado el protocolo IPv4 e IPv6 por DHCP, podemos ver la configuración del protocolo ejecutando el siguiente comando en el equipo “Cliente-Ubuntu”:

```

# ifconfig -a

eth0  Link encap: Ethernet direcciÃ³n HW 02:fd:00:00:02:01
    Direc.inet: 10.0.1.3 Difus.: 10.0.1.255  Másc:255.255.255.0
    DirecciÃ³n inet6: 2001:720:41c:4001:fd:ff:fe00:201/64 Alcance:Global

```

```

Dirección inet6: fe80::fd:ff:fe00:201/64 Alcance:Enlace
ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Mátrica:1
Paquetes RX:18111 errores:0 perdidos:0 overruns:0 frame:0
Paquetes TX:1114 errores:0 perdidos:0 overruns:0 carrier:0
colisiones:0 long.colaTX:1000
Bytes RX: 1914330 (1.9 MB) TX bytes:190460 (190.4 KB)
Interrupción:10 Dirección base: 0x8000
lo Link encap:Bucle local
Direc. inet:127.0.0.1 Másc:255.0.0.0
Dirección inet6: ::1/128 Alcance:Anfitrión
ACTIVO BUCLE FUNCIONANDO MTU:16436 Mátrica:1
Paquetes RX:6 errores:0 perdidos:0 overruns:0 frame:0
Paquetes TX:6 errores:0 perdidos:0 overruns:0 carrier:0
colisiones:0 long.colaTX:0
Bytes RX:534 (534.0 B) TX bytes:534 (534.0 B)

```

Al igual que antes, la asignación de la dirección IPv4 por parte del servidor al cliente se puede comprobar viendo las asignaciones del servidor dhcp almacenadas en el fichero **/var/lib/dhcp3/dhcpd.leases**:

```

# The format of this file is documented in the dhcpd.leases (5) manual page.
# This lease file was written by isc-dhcp-V3.1.3
lease 10.0.1.2 {
starts 0 2011/07/03 12:05:04;
ends 0 2011/07/03 12:15:04;
cltt 0 2011/07/03 12:05:04;
binding state active;
next binding state free;
hardware ethernet 02:fd:00:00:01:01;
uid "\001\002\375\000\000\001\001";
client-hostname "Xp-1";
}
lease 10.0.1.3 {
starts 0 2011/07/03 12:06:27;
ends 0 2011/07/03 12:16:27;
cltt 0 2011/07/03 12:06:27;
binding state active;
next binding state free;
hardware ethernet 02:fd:00:00:02:01;
client-hostname "Cliente-Ubuntu";
}

```

```
}
```

También podemos comprobar la conectividad IPv4 del equipo haciendo una comprobación como la siguiente:

```
# ping 10.0.3.3
PING 10.0.3.3 (10.0.3.3) 56(84) bytes of data.
64 bytes from 10.0.3.3: icmp_seq=1 ttl=63 time=0.835 ms
64 bytes from 10.0.3.3: icmp_seq=2 ttl=63 time=0.834 ms
64 bytes from 10.0.3.3: icmp_seq=3 ttl=63 time=0.849 ms
64 bytes from 10.0.3.3: icmp_seq=4 ttl=63 time=0.805 ms
^C
--- 10.0.3.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.805/0.830/0.849/0.038 ms
```

Para comprobar que el servicio RADVD instalado en “**Gateway**” asigna el router por defecto IPv6 al cliente ubuntu, podemos comprobar la conectividad IPv6 haciendo una comprobación como la siguiente:

```
# ping6 2001:720:41c:4004::2
PING 2001:720:41c:4004::2(2001:720:41c:4004::2) 56 data bytes
64 bytes from 2001:720:41c:4004::2: icmp_seq=1 ttl=63 time=3.13 ms
64 bytes from 2001:720:41c:4004::2: icmp_seq=2 ttl=63 time=0.832 ms
64 bytes from 2001:720:41c:4004::2: icmp_seq=3 ttl=63 time=0.864 ms
64 bytes from 2001:720:41c:4004::2: icmp_seq=4 ttl=63 time=0.825 ms
64 bytes from 2001:720:41c:4004::2: icmp_seq=5 ttl=63 time=2.50 ms
^C
--- 2001:720:41c:4004::2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.825/1.633/3.137/0.991 ms
```

5.3.4 Firewall

Introducción

Otro de los servicios que actualmente es necesario implantar es el firewall. Un firewall es un dispositivo que filtra el tráfico entre redes.

Con el firewall podemos proteger a nuestros equipos de ataques procedentes desde el exterior y desde el interior de nuestras redes. Para que sea efectivo, el filtrado de paquetes debe hacerse tanto en el protocolo IPv4 como en IPv6.

Instalación

Dada la topología de red utilizada para la simulación, lo más lógico es filtrar los paquetes IPv4 e IPv6 en el “**Gateway**” ya que es el equipo por el que pasan todos los paquetes procedentes o dirigidos a los servidores o clientes utilizados.

El sistema Ubuntu dispone de un conjunto de herramientas o comandos que permiten enviar mensajes al kernel para manejar los paquetes TCP/IP de acuerdo a como se desee. Este conjunto de herramientas se llaman IPTABLES para manejar paquetes IPv4 e IP6TABLES para manejar paquetes IPv6.

Por lo tanto, IPTABLES es un sistema de firewall vinculado al kernel de Linux de forma que permite indicarle algunas cosas que debe hacer con cada paquete, esto se hace en base a las características de un paquete en particular.

Como el equipo que se va a utilizar como firewall es un sistema Ubuntu, no es necesario instalar nada adicional.

Configuración

La configuración de IPTABLES e IP6TABLES dependerá de lo que queremos que filtre.

Para configurar las reglas hemos utilizado un programa llamado FWBuilder [16] que es una aplicación que actualmente soporta iptables, ipfilter, OpenBSD PF, y Cisco PIX. Una de sus principales fortalezas es su capacidad para simplificar la administración de múltiples firewalls usando una base de datos de objetos de red central y permitiendo fácilmente instalar la misma política en todos los cortafuegos simultáneamente, sin tener en cuenta la plataforma. FWBuilder también te permite obtener una representación visual de las políticas de seguridad.

FWBuilder permite crear políticas IPv4 (ignorando cualquier regla IPv6), IPv6 (ignorando cualquier regla IPv4) o mixtas.

FWBuilder genera un fichero con todas las reglas para configurar el firewall. Este fichero lo llamamos “**Gateway.escuela.upm.es.fw**” y lo almacenamos en el directorio **/etc** y configuramos el sistema para que cada vez que la máquina se reinicie, se ejecute.

Para ejecutar las reglas de filtrado debemos hacer lo siguiente:

```
# /etc/Gateway.escuela.upm.es.fw
Running prolog script
Activating firewall script generated Fri Jun 24 12:57:53 2011 by rafa
```

```
Rule 0 (eth5)
Rule 1 (lo)
Rule 2 (global)
Rule 3 (global)
Rule 4 (global)
Rule 5 (global)
Rule 6 (global)
Rule 7 (global)
Rule 8 (global)
Rule 9 (global)
Rule 10 (global)
Rule 11 (global)
Rule 12 (global)
Rule 13 (global)
Rule 14 (global)
Rule 2 (global)
Rule 3 (global)
Rule 4 (global)
Rule 5 (global)
Rule 6 (global)
Rule 7 (global)
Rule 8 (global)
Rule 10 (global)
Rule 11 (global)
Rule 12 (global)
Rule 13 (global)
Rule 14 (global)
Running epilog script
```

Pruebas

Para probar el funcionamiento del firewall simplemente tenemos que ver la diferencia del comportamiento de “**Gateway**” cuando tiene activadas y desactivas las reglas de filtrado del firewall.

Para ver si qué reglas en IPv4 están activadas:

```
# iptables -L
```

Para ver si qué reglas en IPv6 están activadas:

6 Conclusiones

La necesidad de migrar hacia el nuevo protocolo de Internet IPv6 es cada vez más inminente, especialmente en entornos educativos tecnológicos, si éstos no quieren frenar su avance en investigación y docencia. Las Escuelas de la UPM deben afrontar este reto y los Servicios Informáticos de los diferentes Centros deben estar preparados para llevarlo a cabo.

Gracias a las herramientas de virtualización es posible realizar una simulación del posible escenario con el que el personal de los diferentes centros de cálculo de la Universidad se va a encontrar a corto o medio plazo. Estas técnicas de virtualización, utilizadas de forma conveniente, pueden suponer un importante ahorro de costes en la implementación del sistema, pues permiten identificar y resolver los problemas de la transición de IPv4 a IPv6 en un entorno prácticamente idéntico al real. En este escenario los técnicos pueden trabajar con IPv6, diseñar sus estrategias de transición y hasta configurar los servicios con ambos protocolos, para que en el momento del cambio real al nuevo protocolo, todo haya sido analizado y estudiado y de esta manera los errores que se puedan producir sean mínimos.

VNX/VNUML es una herramienta de virtualización de propósito general flexible y que simplifica el trabajo del usuario ofreciéndole una gran potencia pero sin tener que enfrentarse a los detalles complejos de creación de máquinas y redes virtuales. De esta forma, el usuario puede concentrarse en la definición de la simulación, que es lo realmente importante desde su punto de vista.

El trabajo realizado en este Trabajo Fin de Máster junto con el realizado en el Trabajo Fin de Máster de Gloria Martín Martín, refleja los principales pasos a seguir en cualquier Escuela de la UPM para conseguir que los servicios básicos demandados por la Comunidad Universitaria, funcionen tanto en IPv4 como en IPv6 durante el periodo de transición al nuevo protocolo, utilizando en ellos una configuración de doble pila de protocolos. También se refleja en estos dos Trabajos un posible mecanismo para comunicar las redes sólo-IPv6, que se puedan crear en las Escuelas con los sistemas IPv4 que probablemente todavía perdurarán durante un tiempo.

Tras el trabajo realizado, los autores de estos dos Trabajos Fin de Máster podemos concluir que el uso de VNX/VNUML para simular la transición de IPv4 a IPv6 resulta muy adecuado ya que es una herramienta que permite, sin necesidad de una gran

inversión económica, poder conocer cómo funciona IPv6, configurar los servicios que se deben ofrecer, resolver los problemas con los que nos vamos a encontrar cuando se lleve a cabo el despliegue y tener definido un escenario prácticamente idéntico al que vamos a encontrarnos en el momento de la transición.

7 Referencias

- [1] L. SEAWRIGHT Y R. MACKINNON, "VM/370: A STUDY OF MULTIPLICITY AND USEFULNESS", IBM SYSTEMS JOURNAL, 18(1), 1979.
- [2] WWW.XEN.ORG.
- [3] www.vmware.com
- [4] J. DIKE, "USER MODE LINUX", PROC. 5TH ANNUAL LINUX SHOWCASE & CONF., OAKLAND CA, 2001.
- [5] HTTP://WWW.GNS3.NET/
- [6] HTTP://WWW.NETKIT.ORG/
- [7] HTTP://WWW.MARIONNET.ORG/
- [8] "VNX HOME PAGE", HTTP://WWW.DIT.UPM.ES/VNX
- [9] T. BRAY, J. PAOLI, C. M. SPERBERG-MCQUEEN, E. MALER, "EXTENSIBLE MARKUP LANGUAGE (XML) 1.0 (SECOND EDITION)", W3C RECOMMENDATION, OCTUBRE 2000.
- [10] "THE PERL DIRECTORY", <http://www.perl.org>.
- [11] "THE LINUX KERNEL ARCHIVES", <http://www.kernel.org/>
- [12] "802.1Q: VIRTUAL LANs", IEEE 802.1Q WORKING GROUP, IEEE, 2001.
- [13] TECNOLOGÍA XML EN LA HERRAMIENTA DESIMULACIÓN DE REDES VNUML, FERMÍN GALÁN, 2004
- [14] UPM UNIVERSITY MIGRATION TO IPV6, T. DE MIGUEL Y D. FERNÁNDEZ, MAR 2003.
- [15] DIBBLER
- [16] FWBUILDER