ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE INGENIERÍA

ESTUDIO Y ANÁLISIS PARA LA MIGRACIÓN DE IPv4 A IPv6 PARA UN DISTRIBUIDOR DE INTERNET

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

JUAN CARLOS OLEAS CASTELO

DIRECTOR: ING. FERNANDO FLORES

Quito, Noviembre 2001

DECLARACIÓN

Yo, Juan Carlos Oleas Castelo, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley, Reglamento de Propiedad Intelectual y por la normatividad institucional vigente.

Juan Carlos Oleas Castelo

AGRADECIMIENTOS

A mi esposa Marjorie, quien con su amor y entrega me ha dado el aliento y el soporte necesarios para llegar a esta meta; A mis padres, por su cariño y apoyo permanentes

Mi especial agradecimiento al Ingeniero Fernando Flores, quien con su profesionalismo me ha brindado una ayuda leal y desinteresada en la realización de este proyecto

Juan Carlos Oleas

DEDICATORIA

A mi Padre Dios quien siempre ha guiado e iluminado mis caminos, a mi esposa Marjorie y mi hijo Juan Josué por ser los pilares fundamentales de mi vida.

Juan Carlos Oleas

CONTENIDO

Pág.

ÍNDICE DE GRÁFICOS

CAPÍTULO 1

CAPÍTULO 2

CAPÍTULO 3

CAPÍTULO 4

PRESENTACIÓN

La red Internet, basada en un diseño de inicios de los años 80, ha experimentado un crecimiento en la historia de las telecomunicaciones tanto en número de usuarios conectados como en aplicaciones y servicios disponibles.

Aunque algunos de los nuevos servicios que encontramos en la Internet se podrían encuadrar dentro del marco aún difuso de las llamadas autopistas de la información por su naturaleza multimedia, la disponibilidad creciente de infraestructura de transmisión a alta velocidad y por su ámbito de difusión global, es bien cierto que han aparecido deficiencias en los aspectos administrativos y de seguridad así como carencias de cara a la prestación de los servicios avanzados que apuntan en el horizonte.

Para remediar estos males, los cuerpos técnicos de la Internet impulsaron un debate bajo el lema de IP Next Generation (IPng) que ha culminado con la especificación de un nuevo protocolo IP, sucesor del actual IPv4, y conocido formalmente como la versión 6 del Protocolo Internet o IPv6.

IPv6 es la nueva versión del protocolo IP, el estudio para la aparición de este protocolo se debe a la IETF (Internet Engineering Task Force), y se documentó en el RFC 1752 como "proposed Standard". En el diseño de este nuevo protocolo se ha tenido en cuenta la resolución de problemas y carencias que presentaba la versión anterior; como por ejemplo la ampliación del espacio de direccionamiento, funcionamiento óptimo en redes de gran rendimiento (ATM), temas de seguridad, calidad del servicio (QoS), nuevo tipo de direccionamiento.

El objetivo de este proyecto de titulación es analizar los requerimientos necesarios para una migración del protocolo IPv4 al IPv6 aplicados a un Distribuidor de Internet, así como también pretende ser una guía de consulta del nuevo protocolo IP versión 6 y de los principales mecanismos de transición hacia éste.

El desarrollo de este proyecto de titulación consiste de cinco capítulos mediante los cuales se pretende dar una idea general sobre el nuevo protocolo IP, los mecanismos de transición hacia éste y aplicarlo al Backbone de un Distribuidor de Internet.

En el primer capítulo se da una introducción del actual protocolo IP, y se proporcionan algunas características tales como direccionamiento, el datagrama Internet, fragmentación, y protocolos que interactúan con IPv4.

La primera parte del segundo capítulo trata sobre las principales características del nuevo protocolo IP tales como el nuevo direccionamiento, encaminamiento, y sus múltiples opciones. En la segunda parte, se expone sobre las seguridades y el nuevo DNS para IPv6.

En el tercer capítulo se estudia a profundidad los diferentes mecanismos de transición de IPv4 a IPv6 tales como capa IP dual, traducción de cabeceras, tunneling en sus diferentes configuraciones y el mecanismo de transición para el DNS.

En el cuarto capítulo se describe primeramente el funcionamiento del Backbone del Distribuidor de Internet; para luego analizar los requerimientos necesarios de los equipos que conforman el Backbone para la migración de IPv4 a IPv6. Por último se describen las características de los dispositivos de varios fabricantes que permiten manejar el nuevo protocolo IPv6.

Para finalizar el proyecto, el quinto capítulo contiene las conclusiones y recomendaciones relacionadas al desarrollo de todo el trabajo.

CAPITULO 1

PROTOCOLO IP
(versión 4)

CAPITULO I

PROTOCOLO IP (versión 4)

1.1 INTRODUCCIÓN

El protocolo por el cual se mantiene unida la Internet es el protocolo de capa de red IP (Internet Protocol, Protocolo Internet). A diferencia de la mayoría de los protocolos de capa de red, éste fue diseñado para interconexión de redes. Su trabajo es proporcionar un medio mejor para el transporte de datos del origen al destino, sin importar si las máquinas están en la misma red o si hay otras redes entre ellas. IP se ocupa de la transmisión de bloques de datos, llamados datagramas de origen a destino, donde orígenes y destinos son hosts identificados por direcciones de una longitud fija. IP también se encarga de la fragmentación y reensamblado de datagramas, si éstos fueran necesarios.

El protocolo IP implementa dos funciones básicas: Direccionamiento y fragmentación.

El módulo Internet usa las direcciones contenidas en la cabecera de los datagramas para transportarlos a sus destinos. Así mismo, existen otros campos en la cabecera que permiten gestionar la fragmentación y posterior reensamblado de datagramas, para poderlos transmitir a través de redes que trabajan con tamaños de paquetes pequeños.

El módulo Internet reside en cada *host* integrado en la Internet, y en cada *router* que interconecta redes, además sigue reglas comunes para interpretar las direcciones y para realizar la fragmentación/reensamblado de datagramas. Adicionalmente, estos módulos (especialmente en los *routers)* están provistos de mecanismos para tomar decisiones sobre el enrutamiento de los datagramas.

Esta faceta del protocolo IP presenta la ventaja de facilitar la interconexión de subredes de diferente tecnología y es sin duda una de las claves del éxito de la Internet como red de redes en contraposición con otras tecnologías orientadas a la conexión como X.25 o ATM (Modo de transferencia asincrónica que transporta eficientemente todo tipo de tráfico sobre una simple red), basadas en el concepto de circuitos virtuales o canales fiables que asigna la red para la comunicación entre extremos de manera ordenada e íntegra.

1.2 DIRECCIONAMIENTO IPv4[1]

Las direcciones son un ingrediente esencial el cual permite a IP ocultar los detalles de las redes físicas, y hace que la red de redes parezca una sola entidad uniforme. Cada máquina en Internet, tiene asignado un número denominado dirección Internet o dirección IP. Este número es asignado de tal forma que se consigue una gran eficiencia al encaminar paquetes, ya que codifica la información de la red a la que está conectado, además de la identificación del *host* en concreto.

Cada dirección Internet tiene una longitud fija de 32 bits, los bits de las direcciones IP de todos los host de una red determinada comparten un prefijo común. Conceptualmente, cada dirección IP es una pareja formada por identidad de red-identidad de host o computador, donde la identidad de red representa a la red, e identidad de host, a un host determinado dentro de esa red.

Para que exista una flexibilidad en la asignación de direcciones, existen tres formatos básicos de representación de direcciones. La elección de uno de estos formatos dependerá del tamaño de la red. Además de los tres formatos básicos, existe uno para *multicasting* (se refiere a tareas de multidifusión), utilizado para envío de mensajes a un grupo de hosts, y otro reservado para uso futuro.

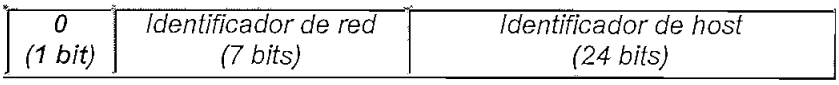
---

[1] Redes Globales de Información con Internet y TCP/IP. Principios básicos, protocolos y arquitectura, Douglas E. Comer, Prentice Hall, Tercera edición 1996.

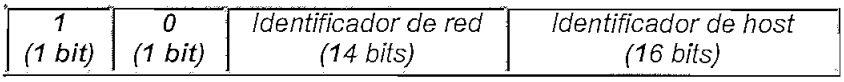
Los primeros bits identifican la clase de dirección IP, que va seguido de un prefijo de identificación de red, y seguido de un identificador de host. La clase D se usa para transmitir un mismo mensaje a un grupo de hosts determinado.

La estructura de los diferentes formatos es la que sigue :

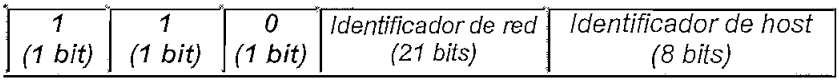
Clase A :



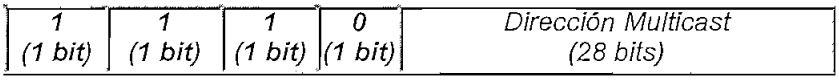
Clase B :



Clase C :



Clase D :



Clase E :

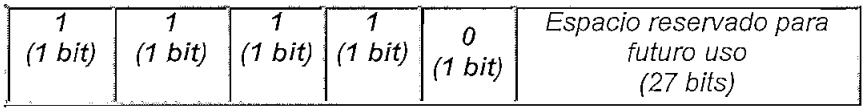


Fig 1.1 Estructura de los formatos de direccionamiento IP[2]

La clase A se usa para grandes redes que tengan más de 2^{16} (65536) hosts, asignan 7 bits al identificador de red y 24 bits al identificador de host. La clase B se usa para redes de tamaño mediano, entre 2^{8} (256) y 2^{16} hosts. Finalmente, la clase C corresponde a redes con menos de 256 hosts. El cuarto tipo, el D, se dedica a tareas de *multicasting.* Nótese que las direcciones IP se han definido de tal forma que es posible extraer rápidamente los campos de red y host.

[2]Redes Globales de Información con Internet y TCP/IP. Principios básicos, protocolos y arquitectura, Douglas E. Comer, Prentice Hall, Tercera edición 1996.

Para asegurar que la parte de identificación de red de una dirección Internet es única, todas las direcciones son asignadas por una autoridad central, el Centro de Información de Red (NIC, Network Information Center).

Esta autoridad central tan sólo asigna el prefijo de red de la dirección y delega la responsabilidad de asignar las direcciones de host individuales a la organización solicitante. A las redes de área local con pocos ordenadores (menos de 255) se le asigna direcciones de la clase C, pues se espera que surjan un gran número de ellas. A redes muy grandes, como ARPANET, se les asigna la clase A, ya que se espera que no surjan demasiadas.

A la hora de trabajar con direcciones IP, usamos la notación decimal, la dirección expresada de esta forma vendrá dada por cuatro enteros positivos separados por puntos, donde cada entero se corresponde con el valor de un octeto de la dirección IP. Por lo tanto, para entender la relación entre los tipos de direcciones IP y los números decimales con puntos se resumen el rango de valores para cada tipo.

TIPO	Dirección más baja	Dirección más alta
A	*0.1.0.0*	*126.0.0.0*
B	*128.0.0.0*	*191.255.0.0*
C	192.0.0.0	*223.255.255.0*
D	*224.0.0.0*	*239.255.255.255*
E	240.0.0.0	247.255.255.255

Fig 1.2 Rango de valores decimales correspondientes a cada dirección IP

Según lo indicado, una dirección IP identifica a un host, pero esto no es estrictamente cierto. Por ejemplo, si un *router* está conectado a dos redes diferentes, no podemos asignarle una dirección IP única, ya que las dos redes tienen su propia dirección de red. En este caso, hay que asignar una dirección

diferente según la conexión, con lo que la dirección IP no especificaría una máquina en particular, sino una conexión a una red.

Según esto, un router que conecte 'n' redes tendrá 'n' diferentes direcciones IP, según la conexión establecida. Adicionalmente si un host se mueve de una red a otra, su dirección IP deberá cambiar según la red en la que se encuentre (como ejemplo, podemos imaginar un ordenador portátil).

Como debilidades del protocolo podemos indicar que si una red crece por encima de lo que su clase le permite direccionar (p.e. Una red de clase C que crezca por encima de los 255 host) deberá cambiar todas sus direcciones a la clase B, proceso muy costoso.

1.2.1 DIRECCIONES ESPECIALES[3]

Existen algunas combinaciones de 0's y 1's que no se asignan como dirección IP, sino que tienen asociado un significado especial. Los dos primeros casos, indicados en la fig.1.3 sólo pueden ser usados al arrancar el sistema (en máquinas sin unidad de almacenamiento fijo) y nunca se usan como una dirección de destino válida. En cualquier caso, sólo se usan de forma temporal mientras el host *aprende* su dirección IP.

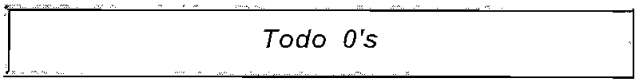
El tercer caso es la denominada dirección de multidifusión de red local, o dirección de multidifusión limitada, que permite difundir un mensaje a toda la red local independientemente de su dirección IP asignada. Un host puede usar esta dirección como parte de un procedimiento de comienzo antes de conocer su dirección IP o la dirección IP de su red.

La dirección de multidifusión dirigida a una red nos permite enviar un mensaje a todas las estaciones situadas en una red determinada. Es una herramienta muy potente, ya que permite enviar un sólo paquete que será difundido en toda la red.

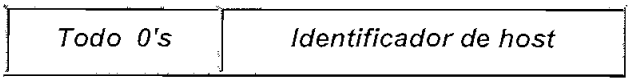
---

[3]Redes Globales de Información con Internet y TCP/IP. Principios básicos, protocolos y arquitectura, Douglas E. Comer, Prentice Hall, Tercera edición 1996.

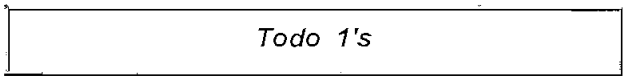
Esta dirección se usa de forma restringida, ya que supone una gran carga de trabajo en redes grandes.



Identifica al propio host.

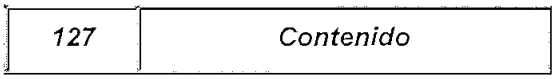


Identifica al host en su red.



Multidifusión limitada en la propia red.

Multidifusión a todos los hosts de la red indicada



Bucle local.

Fig 1.3 Direcciones especiales[4]

La dirección de bucle local está diseñada para pruebas y comunicación entre procesos en la máquina local. Si un programa envía un mensaje a esta dirección, el módulo Internet le devolverá los datos sin enviar nada a la red. De hecho, nunca existirá en la red un paquete de este tipo, ya no es una dirección de red válida.

[4] Redes Globales de Información con Internet y TCP/IP. Douglas E. Comer, Prentice Hall, Tercera edición 1996.

1.2.2 SUBREDES[5]

En el direccionamiento IP a cada red física se le asigna una única dirección de red, los hosts de esa red llevan su dirección de red incluida en su dirección individual.

Este esquema de direccionamiento tiene un fallo: el crecimiento exponencial de Internet. Cuando el protocolo IP fue diseñado, nadie imaginó que pudiera haber cientos de miles de pequeñas redes de ordenadores personales. Al existir tantas redes, aparte del problema administrativo de asignar direcciones a todas ellas, existe el problema de que las tablas de encaminamiento de los *routers* son excesivamente largas, y el ancho de banda utilizado para transmitir esas tablas es alta.

Para solucionar esto, se dividió una red en varias partes para uso interno a estas partes se les llama subredes, lo que permite disminuir el número de direcciones de red asignadas sin alterar el esquema de direccionamiento original. Para conseguir esto, hay que hacer que un mismo prefijo de red IP pueda ser compartido por múltiples redes físicas.

Para conseguir este objetivo deberán modificarse los procedimientos de encaminamiento y todas las máquinas que se conectan a esas redes deben entender las convenciones usadas.

Para asignar una única dirección IP a varias redes físicas, se usa la *máscara de subred*, conceptualmente, el añadir subredes sólo varía la interpretación de las direcciones IP ligeramente. En lugar de dividir la dirección IP de 32 bits en un prefijo de red y un sufijo de host, lo que se hace es mantener el prefijo de red original, pero en lugar de una parte de host, lo que se tiene es una parte local, que puede ser asignada libremente en la red local.

---

[5] Redes Globales de Información con Internet y TCP/IP. Principios básicos, protocolos y arquitectura, Douglas E. Comer, Prentice Hall, Tercera edición 1996.

El resultado es una forma de direccionamiento jerárquico que conlleva a un encaminamiento jerárquico. La ventaja de usar direccionamiento jerárquico es que permite el crecimiento con facilidad, y un *router* no necesita conocer con tanto detalle los destinos remotos como los cercanos. Una desventaja es la dificultad de establecer el sistema, aún cambiarlo una vez establecido.

Para permitir la máxima flexibilidad al dividir las direcciones de subredes, se permite que cada red física pueda escoger independientemente su interpretación propia de subred. Una vez escogida, todas las máquinas en esta red deberán respetar su estructura.

El estándar IP para subredes establece que para cada red física en una localización que utilice subredes hay que escoger una *máscara de subred* de 32 bits. De esta forma, la parte local asociada con el identificador de host se puede dividir en dos, una asociada con el identificador de subred, y otra con el host en particular. En la máscara adquieren valor 1 los bits situados en las posiciones para la indicación de la clase de dirección y para el prefijo de red. Y dentro de la parte local, adquieren valor 1 los bits destinados a identificar la subred.

De esta forma, por ejemplo, una máscara 255.255.255.0 sobre una dirección de clase B determina que se pueden codificar 256 subredes, y sobre cada una de ellas 256 host.

1.3 EL DATAGRAMA INTERNET[6]

Un datagrama IP está formado por una cabecera y un campo de datos. La cabecera tiene una parte fija de 20 bytes y una parte opcional de longitud variable. Un datagrama es la unidad básica de transferencia entre la Internet.

La estructura de un datagrama Internet es la siguiente:

---

[6] http://www.everex.es/ip.htm

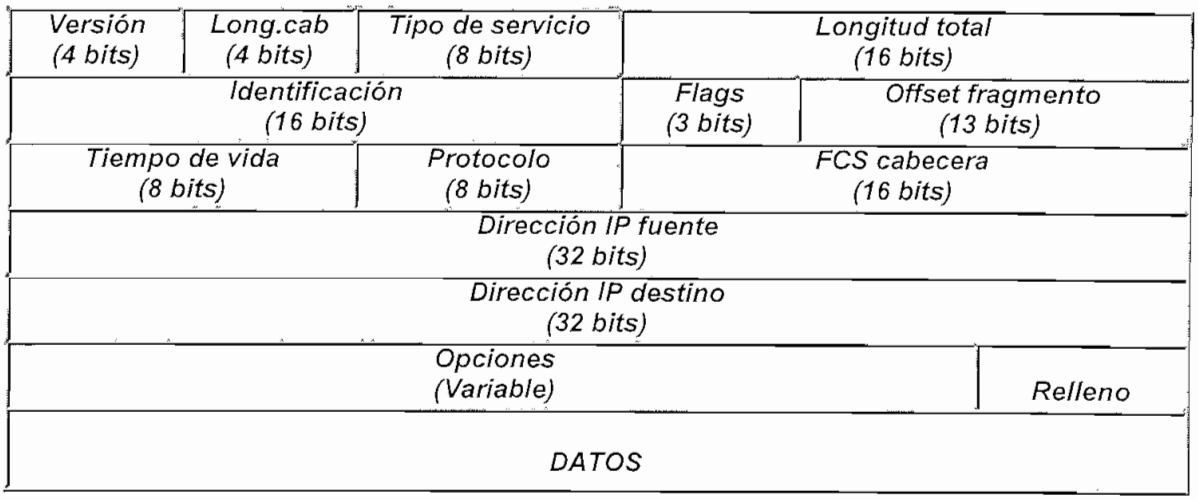


Fig 1.4 Datagrama Internet[7]

A continuación describiremos cada uno de los campos:

1.3.1 VERSIÓN

Este campo ocupa 4 bits, e indica el tipo de formato de datagrama. Para el formato descrito, su valor es 4 (IP versión 4).

1.3.2 LONGITUD DE LA CABECERA

Este campo ocupa 4 bits, y especifica la longitud de la cabecera medida en palabras de 32 bits, el mínimo valor posible para una cabecera correcta es 5 (5x32, 160 bits), ya que el campo de opciones puede estar presente o no. El valor máximo de este campo de 4 bits es 15, lo que limita la cabecera a 60 bytes.

1.3.3 TIPO DE SERVICIO

Este campo ocupa 8 bits, e indica como deberá ser tratado el datagrama. Este campo se divide a su vez en cinco subcampos, de la siguiente forma:

[7] http://www.everex.es/ip.htm

Prioridad (3 bits)	D (1 bit)	T (1 bit)	R (1 bit)	Sin uso (2 bits)

Fig 1.5 Subcampos del tipo de servicio[8]

Los 3 bits de prioridad, con valores comprendidos entre cero (prioridad normal) y siete (control de red), permiten al remitente indicar la importancia del datagrama. Aunque la mayor parte del software y de los routers no usa este campo, es un concepto importante porque permite que en un momento determinado los comandos de control tengan prioridad sobre los datos. Por ejemplo, sin este campo sería imposible implementar algoritmos de control de congestión que no se vieran afectados por la congestión que están intentando controlar.

Los bits D, T y R especifican el tipo de transporte que el datagrama solicita. Si están activos, sus significados son:

- D activado: El datagrama solicita bajo retardo
- T activado: El datagrama solicita alta capacidad
- R activado: El datagrama solicita alta confiabilidad.

Es posible que en uno o varios nodos del camino no exista alguna de las facilidades solicitadas, así, estos bits son más una ayuda a los algoritmos de encaminamiento que una petición de servicio.

Así, si existen varios caminos disponibles a un destino determinado, el algoritmo de encaminamiento usará los bits del campo tipo de servicio para determinar, en función del hardware de red subyacente, el puerto de salida por donde enviar el datagrama.

[8] http://www.everex.es/ip.htm

1.3.4 LONGITUD DEL DATAGRAMA[9,10]

Este campo ocupa 16 bits, e indica la longitud total del datagrama, incluyendo la cabecera y los datos, la longitud se indica en octetos. Con esto, se permite especificar una longitud de hasta 65535 octetos, sin embargo, los datagramas largos resultan inmanejables en muchos hosts y redes; el mínimo tamaño que debería aceptar un host es de 576 octetos. Se recomienda que los hosts sólo envíen datagramas de más de 576 octetos y tener la seguridad de que el destinatario podrá aceptarlos.

El tamaño de 576 octetos se elige para permitir un tamaño razonable del bloque de datos para ser transmitido junto con la cabecera. Así, este tamaño permite un tamaño para el bloque de datos de 512 octetos, junto con 64 octetos para la cabecera. El tamaño máximo de una cabecera es de 64 octetos, y una cabecera normal es de alrededor de 20 octetos, proporcionando un margen de actuación.

Para que el datagrama se transmita de un nodo a otro de la red, deberá ser transportado en un paquete de la red física subyacente. La idea de transportar un datagrama en una trama de red se denomina *encapsulamiento*. Para la red física subyacente, el datagrama IP es como cualquier mensaje intercambiado entre dos ordenadores, sin que reconozca ni el formato de datagrama ni la dirección de destino IP.

En el caso ideal, todo el datagrama IP alcanzará en una sola trama de red, haciendo que la transmisión sea eficiente. Pero como el datagrama puede atravesar en su camino diferentes tipos de redes físicas, no existe una longitud máxima de datagrama que se ajuste a todas ellas. A la longitud máxima de transferencia de datos por trama de una red física se le conoce como unidad de transferencia máxima (MTU, Maximum Transmision Unit). Por ejemplo, Ethernet limita la trasferencia de datos a 1500 octetos, mientras que FDDI (Fiber

[9] Redes Globales de Información con Internet y TCP/IP. Principios básicos, protocolos y arquitectura, Douglas E. Comer, Prentice Hall, Tercera edición 1996.

[10] http://www.everex.es/ip.htm

Distributed Data Interface, estándar definido por la ANSI para la implementación de una LAN de alta velocidad sobre un anillo dual de fibra óptica) permite aproximadamente 4470 octetos por trama.

Cuando un datagrama se envía por una red con un MTU mayor que su longitud, entonces el datagrama se divide en partes denominadas fragmentos, al proceso se le conoce como fragmentación.

1.3.5 IDENTIFICACIÓN

Este campo ocupa 16 bits, y contiene un número entero que identifica al datagrama. Este número suele asignarse con un contador secuencial en la máquina origen que va asignándolos según nuevos datagramas, es indispensable en el proceso de reensamblado de fragmentos, cuando un datagrama fue fragmentado.

1.3.6 FLAGS[11]

Este campo ocupa 3 bits, e incluye varios flags de control:

- Bit 0: Reservado, debe ser 0
- Bit 1: (DF) 0 = el datagrama puede fragmentarse,
  1 = el datagrama NO puede fragmentarse
- Bit 2: (MF) 0 = es el último fragmento
  1 = existen más fragmentos

El primer bit significativo (bit 1) del campo flags es el de *no fragmentación*, se llama así porque si está activo implica que el datagrama no puede fragmentarse. Este bit resulta útil en casos de pruebas de redes y en algunas aplicaciones

[11] Redes Globales de Información con Internet y TCP/IP. Principios básicos, protocolos y arquitectura, Douglas E. Comer, Prentice Hall, Tercera edición 1996.

especiales donde se necesita que el datagrama llegue sin fragmentar. En el caso de que el router sea incapaz de enviarlo sin fragmentarlo, envía un mensaje de error a la máquina origen.

El bit de menor peso del campo flags (bit 2), es el bit de *más fragmentos*. Este bit es útil para la máquina destino, que permite determinar si ha recibido todos los fragmentos correspondientes a un datagrama. Cuando el bit está en cero, indica que es el último fragmento del datagrama. Así, con este bit y con el campo de offset de fragmento, la máquina puede comprobar si ya ha recibido todos los fragmentos y puede reensamblar el datagrama original. La máquina destino no puede guiarse sólo por el bit de *más fragmentos*, porque es posible que se reciba el último fragmento antes de recibir algún fragmento intermedio, ya que IP no provee un método para que los datagramas lleguen ordenados.

1.3.7 TIEMPO DE VIDA[12]

Este campo ocupa 8 bits, e indica cuanto tiempo, en segundos, está el datagrama autorizado a permanecer en el sistema Internet. La idea es simple: cuando una máquina pone un datagrama en la Internet, le asigna un tiempo máximo de existencia del mismo. Los routers y hosts que van procesando el datagrama deben ir decrementando el campo tiempo de vida, y descartarlo de la Internet cuando el tiempo haya expirado.

Es difícil para los routers estimar el tiempo exacto transcurrido desde que el datagrama salió de la máquina anterior, ya que no conocen el retardo inducido por las redes.

Para solventar este problema, se siguen dos normas:

---

[12] Redes Globales de Información con Internet y TCP/IP. Principios básicos, protocolos y arquitectura, Douglas E. Comer, Prentice Hall, Tercera edición 1996.

1. Cada router por el que pasa el datagrama decrementará en 1 el valor del campo.

2. Para tener en cuenta los casos de routers con gran retardo de tránsito, al llegar el paquete a un router, éste almacenará la hora local de llegada, y en el momento de enviarlo decrementará el valor del campo según el número de segundos que haya estado en el sistema esperando ser enviado.

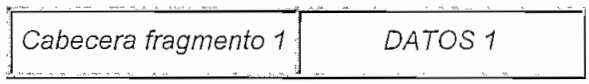
Cuando el campo alcanza el valor cero, el datagrama es descartado y se envía un mensaje de error al origen. La idea del tiempo de vida es interesante porque evita que los datagramas estén eternamente circulando por la red en el caso de que las tablas de encaminamiento estén incorrectas.

1.3.8 OFFSET DEL FRAGMENTO[13]

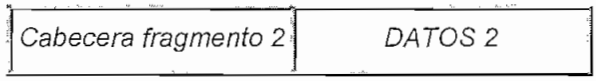
Este campo ocupa 13 bits, y especifica el desplazamiento desde el comienzo del campo de datos del datagrama original hasta el comienzo del campo de datos del fragmento, expresado en múltiplos de 8 octetos. Indica en qué parte del datagrama actual va este fragmento.

Todos los fragmentos excepto el último del datagrama deben tener un múltiplo de 8 bytes, que es la unidad de fragmento elemental, debe haber un máximo de 8192 fragmentos por datagrama, dando una longitud máxima de datagrama de 65536 bytes, uno más que el campo de longitud total, por ejemplo:

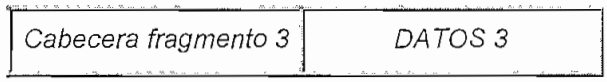
[13] Redes Globales de Información con Internet y TCP/IP. Principios básicos, protocolos y arquitectura, Douglas E. Comer, Prentice Hall, Tercera edición 1996.



Offset : 0 octetos Contenido del campo offset de fragmento: 0



Offset : 600 octetos Contenido del campo offset de fragmento: 75



Offset : 1200 octetos Contenido del campo offset de fragmento: 150

Fig 1.6 Ejemplos de offset

1.3.9 PROTOCOLO[14]

Este campo ocupa 8 bits, e indica cuál fue el protocolo de alto nivel que ha creado los datos que están en el campo *datos.* La asignación de estos valores se hace por una autoridad centralizada (IANA, Institute Assigned Numbers Authority), para que exista acuerdo a través de toda Internet.

1.3.10 FCS CABECERA

Este campo ocupa 16 bits, y asegura la integridad de la cabecera. La máquina origen ejecuta una serie de operaciones matemáticas sobre el conjunto de la cabecera y pone el resultado en este campo. El receptor hará la misma operación y comparará el resultado para asegurarse de que los datos de la cabecera son correctos. Sólo es verificada la cabecera, para no sobrecargar de trabajo a los routers. Al entregarse estos datos sin comprobar, serán los protocolos de alto nivel los que realicen su propio chequeo.

[14] Redes Globales de Información con Internet y TCP/IP. Principios básicos, protocolos y arquitectura, Douglas E. Comer, Prentice Hall, Tercera edición 1996.

1.3.11 DIRECCIÓN IP ORIGEN

Este campo ocupa 32 bits, e indica la dirección IP de la máquina origen

1.3.12 DIRECCIÓN IP DESTINO

Este campo ocupa 32 bits, e indica la dirección IP de la máquina destino.

1.3.13 OPCIONES[15]

Este campo tiene una longitud variable, y puede estar o no presente en la cabecera del datagrama. Esta opcionalidad se refiere a datagramas en particular, no a la implementación específica, cualquier módulo Internet debe implementar esta funcionalidad, tanto en hosts como en routers.

Cada opción tendrá un campo código de opción, de 1 octeto de longitud, que puede ser suficiente según la opción, si no es así, este campo vendrá seguido de un campo *longitud*, también de un octeto, y de un campo conteniendo los datos específicos de la opción de longitud variable.

La estructura de un campo *código de opción* es la siguiente:

Copia	*Clase de opción*	*Número de opción*
1 bit	*2 bits*	*5 bits*

Fig 1.7 Estructura del campo código de opción[16]

1.3.13.1 Copia

El primer bit del campo es el de *copia*. Cuando este bit está a uno, indica que la opción deberá ser copiada a los diferentes fragmentos en caso de que el

---

[15] Redes Globales de Información con Internet y TCP/IP, Douglas E. Comer, Tercera edición 1996.
[16] Redes Globales de Información con Internet y TCP/IP, Douglas E. Comer, Tercera edición 1996.

datagrama sea fragmentado. Si está a cero, entonces la opción deberá ser copiada sólo en el primer fragmento y no en el resto.

1.3.13.2 Clase de Opción

Indica la clase general de opción y establece una opción específica en ésta clase, las diferentes clases son:

00 Datagrama o control de red
01 Reservado para uso futuro
10 Medida y control de errores
11 Reservado para uso futuro

1.3.13.3 Número de Opción

En la siguiente tabla se muestran las diferentes opciones:

Clase	*Número*	*Longitud*	*Descripción*
0	*0*	*1 octeto*	*Fin de la lista de opciones.*
0	*1*	*1 octeto*	*Sin operación.*
0	*2*	*11 octetos*	*Seguridad y restricciones de acceso.*
0	*3*	*variable*	*Encaminamiento de datagramas por rutas específicas.*
0	*7*	*variable*	*Grabación de ruta.*
0	*9*	*variable*	*Encaminamiento dirigido.*
2	*4*	*variable*	*Grabación de tiempo.*

Fig1.8 Clases de opciones[17]

[17] Redes Globales de Información con Internet y TCP/IP. Principios básicos, protocolos y arquitectura, Douglas E. Comer, Prentice Hall, Tercera edición 1996.

- **Fin de la lista de opciones.**

Indica el fin de la lista de opciones, que no suele coincidir con el tamaño de la cabecera. Se usa al final de todas las opciones. Debe ser copiado en caso de fragmentación.

- **Sin operación.**

Esta opción puede usarse entre dos opciones, por ejemplo, para alinear el comienzo de la siguiente opción a 32 bits. Debe ser copiado en caso de fragmentación.

- **Seguridad y restricciones de acceso.**

Esta opción tiene una longitud de 11 octetos, y se divide en varios campos, indicando nivel de seguridad (desde *desclasificado*, hasta *alto secreto*), restricciones de acceso, etc.

- **Opción de grabación de la ruta.**

Las opciones de ruteo y grabación del tiempo son las más interesantes porque proporcionan una manera de monitorear o controlar la forma en que la red de redes maneja las rutas de los datagramas.

Esta opción permite que el origen cree una tabla de direcciones IP vacía, y obliga a cada router por el que pasa el datagrama a incluir su propia dirección IP en dicha lista. Su formato es el siguiente:

Código (8 bits)	*Longitud (8 bits)*	*Puntero (8 bits)*	*primera dirección (32 bits)*	*segunda dirección (32 bits)*	...	*n-ésima dirección (32 bits)*

Fig 1.9. Formato de opción de grabación de ruta[18].

[18] http://www.everex.es/ip.htm

El campo *código* contiene la clase y número de opción (7 para esta opción en concreto). El campo *longitud* especifica la longitud total ocupada por la opción, incluyendo los tres primeros octetos. El campo *puntero* indica la distancia (offset) al primer campo de dirección libre. A continuación, los campos de dirección contienen la dirección IP de los routers que han ido encaminando el datagrama por la Internet.

Cuando el datagrama llega al destino, la máquina leerá la lista y la analizará, pero sólo si antes se ha acordado entre origen y destino que sea así. El destino no analizará las direcciones sólo porque estas aparezcan con la opción activada.

Un detalle a tener en cuenta es que el nodo origen debe reservar suficiente espacio en la lista para las direcciones de todos los routers que el datagrama encuentre en su camino. Si no es así, no se quedarán grabadas las direcciones de los últimos routers que el datagrama atravesó.

- **Opción de encaminamiento dirigido.**

Esta opción permite al remitente indicar un camino determinado al datagrama a través de la Internet. Es muy útil en caso de pruebas, pues permite dirigir el datagrama a un lugar por el que no pasaría normalmente. Para el usuario final no es muy interesante, tan sólo para quienes conocen la topología de la red. Existen dos tipos de encaminamiento dirigido estricto y aproximado.

En el encaminamiento *estricto*, el datagrama debe seguir exactamente el camino indicado en la lista. Si un router no puede conseguir esa ruta, se genera un error. En el caso de encaminamiento *aproximado*, el datagrama puede encontrar varios routers entre dos direcciones IP de la lista sin que se genere ningún error.

En ambos casos los routers van escribiendo su propia dirección IP en la lista, al igual que en la opción de grabación de ruta.

El formato de esta opción es similar al descrito en la opción *grabación de la ruta*, en este caso el código será el 137, y las direcciones IP corresponderán a las de los routers que el datagrama debe atravesar.

➢ **Grabación del tiempo.**

La opción de grabación del tiempo funciona de forma similar a la de grabación de ruta, añadiendo además el momento en que el datagrama atravesó el router. Cada entrada en la tabla contiene dos partes de 32 bits, una será la dirección del router, y otra, el día y la hora expresada en milisegundos desde medianoche, según la hora universal (meridiano de Greenwich). Si no existe posibilidad de obtener esa hora, el router pone a 1 el bit más alto y escribe el día y la hora local. Por esto, la grabación de hora no debe considerarse como exacta, sino como una estimación.

- Los campos código (en este caso 68), longitud y puntero son similares a los de las opciones comentadas anteriormente. La estructura de la opción es la siguiente :

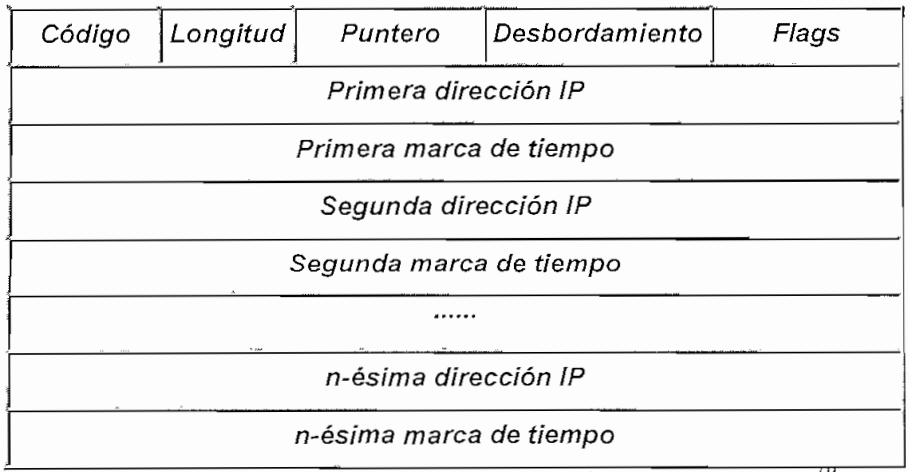


Fig. 1.10 Estructura de la opción grabación del tiempo[19]

[19] http://www.everex.es/ip.htm

- ✓ **Desbordamiento:** Este campo ocupa 4 bits e indica el número de routers que no pudieron insertar sus marcas de tiempo porque no se reservó espacio suficiente.

- ✓ **Flags:** Este campo ocupa 4 bits, e indica como deben los routers insertar exactamente el momento en que el datagrama fue tratado por ellos. Algunos de los valores son:

  0 Grabar sólo el momento. Omitir la dirección IP

  1 Poner la dirección IP antes del momento (este es el formato descrito anteriormente)

  3 Las direcciones IP son indicadas por el remitente. Un router sólo graba el momento si la siguiente dirección IP coincide con la suya.

1.3.14 RELLENO[20]

La cabecera de un datagrama IP está alineada a 32 bits. Este campo se usa para asegurar que sea así. El sobrante hasta conseguir un tamaño múltiplo de 32 (bits), se rellena con 0's.

1.4 FRAGMENTACION[21]

La fragmentación de un datagrama IP es necesaria cuando el tamaño de un datagrama resulta intratable para alguna de las redes que debe atravesar para llegar a su destino.

El campo *identificador* es usado junto con los de dirección origen, dirección destino y protocolo, para identificar fragmentos a reensamblar. El módulo Internet

[20] Redes Globales de Información con Internet y TCP/IP, Douglas E. Comer, Tercera edición 1996.
[21] http://www.everex.es/ip.htm

del origen del paquete debe asignar un identificador único para cada datagrama, que el destino usa para identificar a que datagramas originales pertenecen que fragmento.

El flag *más fragmentos*, está a 1 si el datagrama no es el último fragmento. El campo *offset de fragmento* identifica la localización del fragmento en el datagrama original, indicando el desplazamiento sobre su comienzo.

La estrategia de fragmentación está diseñada para que un datagrama sin fragmentar tenga toda la información relativa a fragmentación a 0's (*más fragmentos* = 0, *offset fragmento* = 0). Si un datagrama es fragmentado, todos sus fragmentos (menos el último) deben estar alineados a 8 octetos (su longitud en bits debe ser múltiplo de 64).

Para fragmentar un datagrama Internet, un módulo Internet crea *n* nuevos datagramas y copia los contenidos de la cabecera a todos ellos. El campo *datos* del datagrama original es dividido en n partes, las cuales deben estar alineadas a 8 octetos. La primera porción de datos se copia en el primer datagrama generado, y se cambia su campo *longitud*, haciéndolo coincidir con la longitud del primer datagrama. El flag *más fragmentos* es puesto a 1. La segunda porción de datos es copiada en el segundo datagrama, se cambia su campo *longitud* y *más fragmentos* de forma similar, y se especifica el desplazamiento en el campo *offset de fragmento.*

Este proceso se repite hasta el último fragmento generado, que tendrá el flag *más fragmentos* a 0, y que no deberá estar alineado a 8 octetos necesariamente.

Para reensamblar los fragmentos, el módulo Internet en el destino, combina los fragmentos que tengan el mismo valor en los campos *identificador, dirección origen, dirección destino* y *protocolo.* La recombinación se hace copiando la parte de datos de cada fragmento en la posición relativa indicada en el campo *offset de fragmento*. El primer fragmento deberá tener el campo *offset de fragmento* a cero, y el último fragmento el flag *más fragmentos* a cero.

1.5 ENCAMINAMIENTO DE DATAGRAMAS IP[22]

Se denomina encaminamiento al proceso de elegir un camino por el cual enviar un paquete, y *router* al sistema encargado de realizar esta decisión.

El propósito del encaminamiento Internet es el de proveer al usuario de una red virtual de envío de datagramas IP sin conexión (diferentes datagramas pueden seguir diferentes caminos), de una forma transparente y sin importar el número o tipo de redes físicas que el datagrama debe atravesar para llegar a su destino. Una Internet está formada por múltiples redes físicas interconectadas por máquinas actuando de *routers*, cada uno de los cuales está unido a dos o más redes físicas. Los hosts, al contrario que los routers, suelen estar conectados a una sola red física.

Tanto los *routers* como los *hosts* participan en el encaminamiento IP, un *host* conectado a varios *routers* decidirá por cual de ellos enviar el datagrama, y un *router* conectado a varias redes decidirá a cual de ellas enviar el datagrama. Un *host* y un *router* pueden coexistir en la misma máquina física, pero el protocolo IP los considera entes totalmente diferentes.

De forma general, podemos dividir el encaminamiento en dos tipos: *Directo* e *indirecto.* El encaminamiento directo es la base del sistema Internet, y consiste en la comunicación de dos hosts enganchados a la misma red física. El remitente deberá encapsular el datagrama en una trama física, mapear la dirección IP en una dirección física (usando el Protocolo de Resolución de Direcciones ARP), y enviar la trama resultante directamente al destinatario. Para que un *host* determine si el destino del paquete pertenece a su misma red, sólo tiene que comparar la parte de la dirección IP que identifica a la red y compararla con la de la propia red.

[22] http://www.everex.es/ip.htm

En la práctica, el encaminamiento directo es la fase final o *entrega* del paquete, pues aunque éste atraviese múltiples redes, al final el último *router* estará en la misma red física que el destino, y usará encaminamiento directo para enviarle el mensaje.

El encaminamiento indirecto se usa cuando el destino no está en la misma red física que el origen, en este caso, el origen enviará el datagrama a un *router*, que determinará a cual de las redes a las que está conectado enviará el datagrama. El encaminamiento indirecto es más difícil que el directo, ya que el remitente debe identificar un router donde enviar el datagrama, el ruteador debe entonces enviar el datagrama hacia su destino final.

En estas situaciones, se utiliza lo que se conoce como *tabla de encaminamiento IP*, donde cada máquina almacena información sobre posibles destinos y como llegar a ellos. Cuando el software de encaminamiento IP necesita transmitir un datagrama, consulta la tabla para decidir dónde enviar el datagrama. En la tabla se almacenan parejas de dirección de red-dirección del siguiente router, indicando al router el camino a seguir hacia la red.

Una técnica usada para que las tablas de enrutamiento no sean demasiado grandes es consolidar múltiples entradas en un caso por defecto. Este tipo de técnica es muy útil en redes pequeñas que tienen una sola conexión con el resto de la Internet, en este caso, el encaminamiento consiste sólo en ver si la estación destino está en la propia red, si no es así, se envía un datagrama al único *router* de comunicación con el exterior.

También pueden almacenarse direcciones de máquinas específicas en la tabla, que sirven para dar más control al administrador y como medida de seguridad, detección, y corrección de errores.

1.6 EL PROTOCOLO DE RESOLUCIÓN DE DIRECCIONES ARP [23]

Dos máquinas que quieren comunicarse en la misma red física, sólo podrán hacerlo si conocen sus direcciones físicas. Existen diversas formas de resolver el problema. Si en la red física pudiera escogerse la numeración de las estaciones (ejemplo: red PRONET 10), entonces podemos hacer que su número sea una función simple de su dirección IP.

Pero en el caso de una red Ethernet, el problema no resulta tan sencillo de resolver. Cada interfaz Ethernet tiene asignada una dirección hardware de 48 bits, así pues, es imposible codificar la dirección hardware en una dirección IP, además, si se sustituye la interfaz Ethernet, cambia la dirección física de la estación.

Para resolver este problema, se diseño el *protocolo de resolución de direcciones* (ARP, Address Resolution Protocol), válido para todas las redes que soportan multidistribución.

La idea es simple, si una máquina A necesita saber la dirección física de una máquina B, envía por multidifusión un paquete especial que pide a la máquina con la dirección IP indicada que responda con su dirección física. Una vez recibida la respuesta, A puede enviar paquetes a B directamente, pues conoce su dirección física.

Debido a que la multidifusión es un recurso costoso (consume recursos de red, ya que todos los receptores deben procesar el paquete enviado), suele evitarse su uso lo más posible. Una de las formas de hacer esto es manteniendo en cada máquina una tabla relacionando direcciones IP con direcciones físicas. Además, como en cada petición ARP se encuentra la dirección IP y la dirección física del remitente, todas las máquinas activas pueden actualizar su tabla con el nuevo dato.

---

[23] Redes Globales de Información con Internet y TCP/IP. Principios básicos, protocolos y *arquitectura*, Douglas E. Comer, Prentice Hall, Tercera edición 1996.

Al enviar un mensaje ARP de una máquina a otra, éste debe viajar en una trama física. Para que la máquina destino identifique la trama como ARP, debe llevar un valor en el campo de tipo de trama que lo identifique como tal. En Ethernet, este valor es 0806h (en hexadecimal). El formato del mensaje ARP no es fijo, sino que depende que hardware de la red.

El formato de un mensaje ARP para Ethernet es el siguiente:

El campo *tipo de hardware (16 bits)* especifica el tipo de interfaz hardware del que se busca la dirección (1 para Ethernet). El campo *tipo de protocolo (16 bits)* especifica el tipo de dirección de protocolo de alto nivel que proporcionó el transmisor, contiene 0800h para la dirección IP.

Los campos de longitud de direcciones física y de protocolo permiten usar ARP con diferentes hardware y protocolos. El campo *operación* nos indica el tipo de operación en concreto, si es una petición ARP o una respuesta a una petición. El resto de los campos indican las direcciones IP y físicas tanto del remitente como del destinatario.

<table>
<tr><td colspan="2">Tipo de hardware</td><td>Tipo de protocolo</td></tr>
<tr><td>Long. dir. física</td><td>long. dir. Protocolo</td><td>Operación</td></tr>
<tr><td colspan="3">Dirección física remitente
(octetos 0 a 3)</td></tr>
<tr><td colspan="2">Dirección física remitente
(octetos 4 a 5)</td><td>dirección IP remitente
(octetos 0 y 1)</td></tr>
<tr><td colspan="2">Dirección IP remitente
(octetos 2 y 3)</td><td>Dirección física destinatario
(octetos 0 y 1)</td></tr>
<tr><td colspan="3">Dirección física destinatario
(octetos 2 a 5)</td></tr>
<tr><td colspan="3">Dirección IP destinatario
(completa, octetos 0 a 3)</td></tr>
</table>

Fig. 1.11 Formato de un mensaje ARP para Ethernet[24]

[24] Redes Globales de Información con Internet y TCP/IP. Principios básicos, protocolos y arquitectura, Douglas E. Comer, Prentice Hall, Tercera edición 1996.

1.6.1 EL PROTOCOLO RARP[25]

El protocolo RARP (Reverse Address Resolution Protocol) es una variación de ARP, que permite a estaciones sin unidad de almacenamiento fija obtener su propia dirección IP.

Cuando una estación sin unidad de almacenamiento arranca, envía a la red un mensaje multidifusión con su dirección física (obtenida directamente del hardware). El servidor de direcciones buscará la dirección física del solicitante y le enviará un mensaje indicándole su dirección IP.

1.7 DNS - Domain Name System[26]

Los primeros sistemas de computadoras forzaban a los usuarios a entender direcciones numéricas para objetos como tablas de sistema y dispositivos periféricos. Los usuarios de las redes, prefieren utilizar nombres pronunciables, más fáciles de recordar, en vez de la dirección IP de las máquinas conectadas a la red.

Inicialmente en Internet, el sistema de nombres escogido era una secuencia de caracteres arbitraria, administrada por el NIC (Network Information Center), que comprobaba la no existencia de otra máquina con ese mismo nombre. Como el número de usuarios se incrementó demasiado, el tener una única autoridad de asignación de nombres no era nada práctico, debido al enorme trabajo administrativo que era mantenerla al día.

La solución hallada, que aún se utiliza, fue el descentralizar el mecanismo de asignación de nombres, delegando la autoridad, y distribuyendo la responsabilidad de asignar la relación entre nombres y direcciones. La definición de asignación entre nombres y direcciones debe estar orientada a la traducción eficiente y que garantice el control autónomo de la asignación de nombres.

---

[25] http://www.everex.es/ip.htm
[26] http://www.everex.es/ip.htm

Pongamos un ejemplo:
nombre_local.nombre_general

donde *nombre_local* sería el nombre administrado por una localización en concreto y *nombre_general* administrado por una autoridad general (nótese que ambos nombres están separados por puntos). Si aparece una nueva localización, la autoridad central incluirá su nombre en la lista de localizaciones válidas y le daría capacidad para administrar todos los grupos de nombres que antecedan al nombre de esa nueva localización (separada por puntos). Los nombres se componen de combinaciones de los 26 caracteres anglosajones (A-Z y a-z), los dígitos (0-9) y el carácter "-". La longitud máxima de nombres de dominios o subdominios es de 63 caracteres y del nombre completo de 255 caracteres.

Así llegamos al punto de tener una estructura jerárquica, subdividiendo el espacio de nombres hasta que éste sea manejable, esto es:

disc.epn.edu.ec

Donde "disc" sería el Departamento de Ingeniería de Sistemas y Comunicaciones, de la Escuela Politécnica Nacional "epn", entidad educativa "edu", "ec" de Ecuador. Podríamos caer en la falacia de que los nombres asignados están relacionados (necesariamente) con la topología de la red o la estructura de las interconexiones físicas.

El mecanismo que implementa una jerarquía de nombres de máquinas en las redes se llama Sistema de Dominio de Nombres (Domain Name System, DNS; a partir de ahora utilizaremos las siglas anglosajonas para referirnos a este sistema, por ser reconocidas internacionalmente).

El DNS especifica la sintaxis de los nombres, y las reglas para delegar autoridad sobre los nombres; además de especificar la implementación de un sistema distribuido que relaciona eficientemente nombres con direcciones.

1.7.1 SINTAXIS DE NOMBRES[27]

La sintaxis de los nombres se compone de nombres de dominios separados por puntos. El nivel más bajo se sitúa a la izquierda (esto facilita comprimir mensajes con múltiples nombres de dominios).

En Internet la máxima autoridad para asignar las direcciones IP y los DNS es la IANA (Internet Assigned Numbers Authority), también es la encargada de delegar el segundo nivel de DNS a la organización IR (Internet Registry) o a registros regionales.

El sistema principal (root) no tiene nombre y es el que en un solo fichero (hosts.txt) tiene los nombres de los host y sus direcciones, éste fichero es accesible vía ftp (RFC 952 y RFC 953).

Ejemplos de códigos de país, según la norma ISO-3166:

Nombre	*Significado*
ec	*Ecuador*
es	*España*
uk	*Inglaterra (United Kindows)*
de	*Alemania (Deuschland)*
us	*EE.UU.. (United States of América)*

Fig 1.12. Ejemplos de códigos de país[28]

El "Top-level domain names" (TLDs, nivel superior del dominio de nombres) se divide en los siguientes DNS:

[27] Redes Globales de Información con Internet y TCP/IP. Principios básicos, protocolos y arquitectura, Douglas E. Comer, Prentice Hall, Tercera edición 1996.
[28] http://www.everex.es/ip.htm

Nombre	Significado
Com	Organizaciones comerciales, se van a establecer subdominios
Edu	Instituciones educativas (registro de 2 a 4 años)
Gov	Instituciones gubernamentales.
Mil	Grupos militares
Net	Principales centros de soporte de red (NICs, nocs...)
Org	Otras organizaciones (diferentes a las anteriores)
Arpa	Dominio ARPANET temporal (obsoleto)
Int	Organizaciones internacionales
Código de país	Cada país (según esquema geográfico)

Fig 1.13. DNS del nivel superior de dominio de nombres (TLD)[29]

Conceptualmente, se permiten dos tipos diferentes de jerarquías: geográfica y organizacional, cada organismo solicita con que tipo de esquema desea tener su nombre (en Internet el esquema geográfico es administrado por organismos generalmente públicos).

Veamos ejemplos de ambos tipos de jerarquías:

ozu.com y ozu.es (dos empresas distintas, nacidas de la separación de Ozu).

La configuración de los host locales (se comenta en el RFC1033) pasa por unas especificaciones del administrador principal, éste le provee de:

- La definición de su zona de actuación.
- El fichero maestro de datos.
- Le actualiza el fichero maestro.

[29] Redes Globales de Información con Internet y TCP/IP. Douglas E. Comer, Tercera edición 1996.

Y el Sistema de Dominio proporciona los métodos estándar de:

- Formatos de recursos de datos.
- Métodos de búsqueda en las BD.
- Métodos para actualizar los datos locales sobre Servidores de Nombres.

En algunos países el segundo nivel de la jerarquía está definida por categorías (AC, CO, GO, RE ...), en otras por políticas geográficas, por ejemplo en EE.UU. es de la forma : nombre-entidad.localidad.estado.us

IBM.Armonk.NY.US

En EE.UU. existe un subdominio en el segundo nivel. El IR se encarga de seleccionar y designar la administración diaria del DNS.

Destaquemos que usando solo la sintaxis de dominio de nombres no se puede distinguir los nombres de subdominios de máquinas individuales.

Nombre	*Significado*
cc	*colegios*
tec	*escuelas técnicas*
state	*agencias estatales de gobierno*
cog	*ayuntamientos*
lib	*bibliotecas*
mus	*museos*
....	*ver RFC-1480 para otros*

Fig. 1.14 Ejemplos de subdominios de segundo nivel en EE.UU[30]

[30] http://www.everex.es/ip.htm

1.7.2 NORMAS PARA NOMBRAR DNS

El registro de un nuevo nombre no implica derechos de marca (C, R, TM, LTD ...), y la responsabilidad es de cada uno al elegir su nombre, asegurándose que no sea una TM (marca registrada).

IANA no se encarga de decidir que es o no un país, estado,etc. por ello la codificación de países la hace mediante la utilización del ISO 3166.

1.7.3 DOMINIO DE NOMBRES PARA CORREO ELECTRÓNICO[31]

Los mensajes de correo son de la forma:

nombre_usuario@nombre_parte_de_dominio

pongamos un ejemplo:

a00444@epn.edu.ec

Donde a00444 sería el nombre del usuario (en este caso algún profesor de la Facultad de Ingeniería Eléctrica de la Escuela Politécnica Nacional), este nombre es configurado por el administrador de la subred, la red local o incluso por un usuario de una máquina conectada directamente a Internet. @ nos indicaría que es una dirección de correo (el sistema de correo utiliza el DNS MX) y epn.edu.ec sería el nombre del dominio (Escuela Politécnica Nacional, Ecuador).

1.7.4 RESOLUCIÓN DE NOMBRES EN DIRECCIONES[32]

El esquema de dominio de nombres incluye un sistema eficiente, seguro, de propósito general y distribuido para relacionar nombres con direcciones. Este sistema está compuesto por una serie de sistemas independientes, pero

[31] http://www.everex.es/ip.htm
[32] http://www.everex.es/ip.htm

cooperativos denominados Servidores de Nombres; cada uno de ellos es un programa que funciona en un servidor y que soporta traducciones de direcciones IP a nombres y viceversa.

El programa cliente, denominado resolvedor de nombres, necesitará usar uno o más servidores al traducir un nombre.

Existen dos tipos de peticiones de resolución:

- **Iterativa**: En la que el servidor, en el caso de que no pueda resolver la dirección por si mismo, mandará un mensaje al remitente diciéndole que no puede resolverla e indicándole la dirección del servidor de nombres al que debe dirigirse para hacerlo.

- **Recurrida**: Donde el servidor de nombres contará con otros servidores hasta hallar la respuesta a la petición y la enviará al remitente.

Para optimizar la búsqueda de nombres en servidores remotos, y para reducir el tráfico en la red, los servidores utilizan la técnica caching, que consiste en:

1. Cuando se recibe respuesta de un servidor remoto de una petición de resolución, se le añade un tiempo de vida (TTL) .

2. Se mantiene durante un cierto tiempo en la memoria del servidor de nombres aquellas parejas nombre-dirección que hayan sido resueltos a petición de algún usuario de la red, junto con su TTL.

3. Antes de enviar una petición a un servidor remoto, buscará en memoria si ya tiene esa dirección resuelta. Si existe en memoria, se la enviará al remitente informándole que pudiera no estar actualizada; enviará también la dirección servidor de nombres remoto, por si le interesa garantizar la veracidad de la información.

1.7.5 LA TRANSMISIÓN DE MENSAJES[33]

La transmisión se produce con octetos. En cada octeto se numera los bits de izquierda a derecha, empezando por 0, siendo éste el de mayor peso. En cuanto a los octetos, éstos se envían en orden de significación. También se puede hacer en ASCII con paridad cero.

1.7.6 FORMATO DEL MENSAJE DE DOMINIO DE NOMBRES[34]

Este mensaje es usado por la aplicación, que debe comunicarse con una máquina y necesita resolver el nombre (que le ha introducido el usuario) para hallar la dirección equivalente, la máquina lo mandará a un servidor de nombres local y éste le contestará con otro mensaje (menor de 512 caracteres):

- **Identificación**: Usado por el remitente para comparar respuestas y preguntas.
- **Parámetro**: Especifica la operación pedida y el código de respuesta (ordenados los bits de izquierda a derecha).
- **Número de pregunta**: Número de entradas en la sección pregunta.
- **Número de respuestas**: Número de entradas en la sección respuestas.
- **Número de autoridad**: Número de entradas en la sección autoridad.
- **Número de añadidos**: Número de entradas en la sección añadidos.
- **Sección de pregunta**: Preguntas sobre las que se solicita respuesta.

[33] Redes Globales de Información con Internet y TCP/IP. Douglas E. Comer, Tercera edición *1996*
[34] http://www.everex.es/ip.htm

Identificación (16 bits)	Parámetro (16 bits)
Número de pregunta	Número de respuestas
Número de autoridad	Número de añadidos
SECCIÓN DE PREGUNTA ... SECCIÓN DE RESPUESTA ... SECCIÓN DE AUTORIDAD ... SECCIÓN DE AÑADIDOS ...	

Fig. 1.15 Formato del mensaje de dominio de nombres. Las secciones de pregunta, respuesta, autoridad y añadidos son de longitud variable[35]

El formato de cada pregunta es:

DOMINIO DE NOMBRES DE LA PREGUNTA (32 bits)	
TIPO PREGUNTA (16 bits)	CLASE PREGUNTA (16 bits)

Fig. 1.16 Formato de la pregunta[36]

bit		significado
0	Operación:	0 Pregunta 1 Respuesta
1-4	Tipo de Pregunta:	0 Estándar 1 Inversa 2 Obsoleta (Terminación 1) 3 Obsoleta (Terminación 2)
5		1 Pregunta de autoridad
6		1 Mensaje Truncado
7		1 Se desea recursión
8		1 Recursión disponible
9-11		Reservado
12-15	Tipo de Respuesta:	0 Sin error 1 error de formato en pregunta 2 Fallo de servidor 3 Nombre no existe

Fig. 1.17 Campo del Parámetro[37]

[35] http://www.everex.es/ip.htm
[36] http://www.everex.es/ip.htm
[37] http://www.everex.es/ip.htm

- **Dominio de nombres de la pregunta**: Contiene el nombre solicitado. El primer octeto indica la longitud de cada etiqueta (en octetos). La última etiqueta es de longitud 0 para indicar el fin del nombre.

- **Tipo pregunta**: Codifica el tipo de solicitud, como por ejemplo: ¿Es una máquina?, ¿Es una dirección de correo?, etc.

- **Clase pregunta**: Nos permite usar este mensaje para otras direcciones que no sean Internet.

- **Sección de respuesta**: El servidor responderá a cada pregunta de la sección anterior, con una respuesta en esta sección. El formato de las secciones Respuesta, Autoridad y Añadidos es:

PETICIÓN DOMINIO DE NOMBRE (32 bits)	
TIPO (16 bits)	*CLASE (16 bits)*
TTL	*LONGITUD PETICION*
DATOS PETICION	

Fig. 1.18 Formato de los campos Respuesta, Autoridad y Añadido[38]

- **Petición dominio de nombre** : Nombre propio (del nodo que pide la resolución).

- **TTL**: Tiempo que debe mantenerse en la memoria del servidor de nombres local, número entero positivo con signo de 32 bits.

- **Longitud de datos recurso**: Número de octetos en la sección datos recurso, 16 bits.

- **Clase**: Especifica la clase de datos.

[38] Redes Globales de Información con Internet y TCP/IP. Douglas E. Comer, Tercera edición 1996

Tipo	Valor	Significado y contenido.
IN	1	Internet.
CS	2	Obsoleta: CSNET.
CH	3	CHAOS.
HS	4	Hesiod.
	255	Ninguna clase.

Fig. 1.19 Tipo, valor y significado de la clase[39]

- **Tipo**: Petición disponible, según:

Tipo	Valor	Significado y contenido
A	1	Dirección de Host: Dirección IP 32 bits.
NS	2	Servidor de Nombres autorizado para el dominio.
MF	4	Obsoleto (fuente de correo).
CNAME	5	Nombre canónico de un dominio.
SOA	6	Inicio de autoridad: Especifica que parte de la jerarquía de nombres esta implementada por un servidor de nombres.
MB	7	Experimental: MDN (Mailbox domain name).
MG	8	Experimental: Miembro de grupo de correo.
MR	9	Experimental: Renombre del MDN.
NULL	10	Experimental: Nulo.
WKS	11	Descripción de servicio conocido bueno.
PTR	12	Nombre del dominio como puntero.
HINFO	13	Nombre de la CPU y del S.O.
MX	15	16 bits prioritarios y nombre del host que actúa como central de correo para ese dominio.
TXT	16	Texto arbitrario: cadena ASCII sin interpretación.
AAAA	28	Dirección de Host: Dirección IPv6 128 bits.
AXFR	252	Petición de transferencia de una zona.
MAILB	253	Petición de campos de correo (MB,MG,MR)
MAILA	254	Obsoleto: Petición resolución de correo.
*	255	Petición de todos los registros.

Fig. 1.20 Tipo de petición[40]

[39] Redes Globales de Información con Internet y TCP/IP. Douglas E. Comer, Tercera edición 1996
[40] Redes Globales de Información con Internet y TCP/IP. Douglas E. Comer, Tercera edición 1996

Los más utilizados son A y MX.

- **Datos recurso**: Aquí se halla la respuesta a la pregunta solicitada.

1.8 DIRECCIONAMIENTO IPv4 EN LA ACTUALIDAD. LA ESTRATEGIA CIDR (CLASSLESS INTER-DOMAIN ROUTING)[41]

En los años 1992-1993 se planteó el problema del espectacular crecimiento de Internet. Los problemas planteados eran:

- Se ve cercano el agotamiento de las direcciones Clase B. Las direcciones Clase C sólo permiten 255 *hosts*, mientras que las de Clase B permiten 65535. En la mayoría de las redes, la Clase C resulta demasiado pequeña, y la B demasiado grande.

- El crecimiento de las tablas de enrutamiento en los *routers* de Internet empieza a hacerlas intratables para el software y hardware existente.

- El espacio de 32 bits para direcciones comienza a resultar escaso.

Los dos primeros problemas son los que se intentan resolver mediante la estrategia CIDR. El tercer problema (el agotamiento del espacio de 32 bits para direcciones), resulta irresoluble en el marco de la versión 4 del protocolo IP.

Durante los años 1992-93 se produce una transición en Internet hacia CIDR, para solucionar los dos primeros problemas.

[41] http://www.everex.es/ip.htm

1.8.1 LA ESTRATEGIA CIDR (CLASSLESS INTER-DOMAIN ROUTING)

La idea básica de la estrategia CIDR es la asignación de uno o más bloques de números de Clase C, y la introducción de una máscara que identifique al conjunto de direcciones.

Además de frenar el agotamiento de las direcciones Clase B, conseguimos una estructura jerárquica que ayuda a frenar el crecimiento de las tablas de enrutamiento.

Un prefijo de red deja de ser fijo, dependiendo de la clase de dirección (A/B/C), y pasa a consistir de una dupla <Dirección IP-Máscara IP>.

Veamos esto con algunos ejemplos:

Imaginemos un sistema que requiere direccionar menos de 1024 *hosts*, a este sistema se le asignarían 4 direcciones Clase C, Por ejemplo, de la 192.24.8.0 a la 192.24.11.0. Para referenciar al conjunto se usaría la dirección 192.24.8.0, con una máscara de dirección 255.255.252.0

Un sistema que requiriera menos de 512 direcciones de *host* tendría asignadas 2 direcciones Clase C, por ejemplo la 192.24.34.0 y la 192.24.35.0. Para referenciar el conjunto tendríamos la dirección 192.24.34.0 con la máscara 255.255.254.0

Como consecuencia de la implantación de la estrategia CIDR, se elimina el concepto de clases de dirección, y tenemos prefijos de red de longitud variable, longitud que viene especificada por la máscara de dirección.

CIDR ha tenido un gran impacto en los sistemas de enrutamiento en Internet. Los prefijos de red de longitud variable han permitido desarrollar múltiples niveles en el sistema de direccionamiento, introduciendo el concepto de *super redes*, esto es, una adecuada asignación y gestión de las máscaras de dirección permite

CAPITULO 2

PROTOCOLO IP (versión 6)

CAPITULO II

PROTOCOLO IP (versión 6)

2.1 INTRODUCCIÓN

La Internet es la red de redes instalada más extensa de TCP/IP (Protocolo de Control de Transmisión/Protocolo Internet), de manera que muchos problemas aparecen en Internet antes de que salgan a la superficie en otras redes TCP/IP. Los investigadores y fundadores del TCP/IP provienen de compañías que utilizan Internet, por lo tanto tienen una motivación para resolver problemas que mejora el servicio y amplía su funcionalidad. Pero ni la Internet ni el conjunto de protocolos TCP/IP son estáticos, nuevos grupos conectan sus redes y descubren nuevas formas de utilizar la tecnología, los investigadores resuelven nuevos problemas y los ingenieros mejoran los mecanismos, es decir la tecnología continúa evolucionando.

La red global de Internet ha tenido varios años de crecimiento exponencial, duplicando su tamaño cada nueve meses o mucho más rápido. A inicios de 1994 un nuevo host aparecía cada 30 segundos, la gente ahora utiliza Internet luego de sus horas de trabajo para actividades comerciales y de entretenimiento, las nuevas aplicaciones que transfieren imágenes y video en tiempo real generan más tráfico que las aplicaciones que transfieren texto.

Las redes de ordenadores se benefician por la disponibilidad de muchas nuevas tecnologías, incluidas ATM, Gigabit Ethernet[42] y LAN Virtuales. La organización de la Internet e Intranets tienen una gran evolución gracias a la adopción del nuevo protocolo IPv6.

[42] Gigabit Erhernet es una extensión a las normas de 10 Mbps y 100 Mbps IEEE 802.3 ofreciendo un gran ancho de banda de 1000 Mbps

2.2 ¿POR QUÉ UNA NUEVA VERSIÓN IP?[43]

IPv6 es una nueva versión de IP diseñada para ser un paso evolutivo de IPv4. Es un incremento natural de IPv4. Puede ser instalado como un software normal mejorado en dispositivos Internet y puede interoperar con el IPv4 actual.

IPv6 está diseñado para correr bien en redes de alto rendimiento como por ejemplo ATM, y al mismo tiempo ser aún eficiente para redes de ancho de banda baja como la radiocomunicación, además; proporciona una plataforma para nuevas funcionalidades de Internet que fueran requeridas en un futuro cercano.

La versión 4 del protocolo de Internet (IPv4) proporciona los mecanismos de comunicación básicos del conjunto TCP/IP, la red global Internet se ha mantenido casi sin cambio desde sus inicios a fines de los años setenta. La antigüedad de la versión 4 muestra que su diseño es flexible y poderoso.

IP permite al usuario la posibilidad de utilizar sus aplicaciones preferidas independientemente de la tecnología usada en capa física.

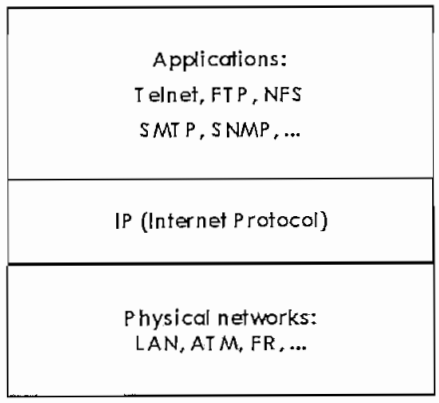


Fig. 2.1 Protocolo IP y su relación con las otras capas[44]

[43] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997
[44] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997

Desde su creación, IPv4 suscitó numerosas discusiones sobre la concepción de la cabecera, el problema más conocido concierne al espacio de direccionamiento. Las direcciones IP están actualmente encapsuladas en 32 bits. Esto permite aproximadamente $4x10^9$ direcciones, lo que parecía suficiente al principio, cuando lo más común era que hubiese un host por campo.

Hoy en día el número de ordenadores personales conectados hace que este número sea demasiado pequeño, sobre todo porque numerosas direcciones están ocupadas por el mecanismo de asignación jerárquica. La generalización de las máquinas conectadas en red va a agravar todavía más este problema.

Otro problema viene dado por el aumento cada vez mayor del tamaño de las tablas de encaminamiento de Internet. El encaminamiento en una red debe ser jerárquico, con una profundidad tan grande como la amplitud de la red. El encaminamiento IP es jerárquico únicamente a tres niveles: red, subred y máquinas o hosts. Los routers de las grandes redes de interconexión deben tener una entrada en sus tablas para todas las redes IP existentes. Este problema es parcialmente resuelto por el CIDR (Classless Inter Domain Routing).

IPv4 no permite indicar de manera práctica el tipo de datos transportados (TOS, Tipo de Servicio en IPv4), y por tanto, la gestión o el nivel de servicio deseado. Esto es necesario particularmente en aplicaciones de tiempo real (como video) y en general para todo tipo de servicios.

El crecimiento es la principal causa que provocó la necesidad de una siguiente generación IP. Actualmente los servicios IPv4 podrían ser llamados el mercado de la computadora, éste ha sido el conductor del crecimiento de Internet. Este mercado ha estado creciendo a una tasa exponencial. El reto para IPv6 es dar una solución que resuelva los problemas de hoy y sea atractivo para mercados emergentes.

IPv6 necesita un nuevo esquema de direccionamiento con las siguientes características:

- Un gran número de bits de modo que el espacio de direccionamiento no se agote.
- Una organización de direccionamiento jerárquica más flexible que no use el concepto de clases, sino únicamente el mecanismo del CIDR.
- Un esquema para la asignación de direccionamientos que reduzca al mínimo el tamaño de las tablas de rutas en los routers e incremente el rendimiento del CIDR.
- Direcciones Globales para la Internet y direcciones locales para las Intranets.

2.3 HISTORIA DE IPv6[45,46]

La estandarización de IPv6 comenzó formalmente en 1992, cuando el IETF (Internet Engineering Task Force, dedicados a la investigación de la evolución de la configuración de Internet), durante la reunión de Boston hizo un llamado para presentar propuestas para IPv6, y muchos grupos de trabajo se crearon.

IPv6 fue recomendado por los directores del área de IPng (Internet Protocol Next genaration) de la Internet Engineering Task Force (IETF) en la reunión de Toronto el 25 de Julio de 1994, y documentado en la RFC 1752 Recomendación para el Protocolo IP Next Generation.

El nombre formal de este protocolo es IPv6, y representa la evolución de muy diferentes propuestas de IETF y grupos de trabajo, además representa tres años de continuo esfuerzo.

Los grupos de trabajo se dedicaron a encontrar una solución al déficit de las direcciones IP de 32 bits, pues éstos no van de acuerdo con la identificación lógica de los hosts y de la estructura jerárquica.

---

[45] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997
[46] http://ao1-unix.gsyc.inf.uc3m.es/~baudoux/intro.htm

Se elaboraron dos grupos de trabajo, el primero dedicado a corregir el sistema de direcciones demasiado rígido, ya que actualmente se pueden conectar unos 16 millones de máquinas por cada clase A, más de 65000 de por cada clase B y 254 por cada clase C. Las clases A y B están casi agotadas. El CIDR permite optimizar las direcciones, pero esto es tan solo transitorio, pues luego de algún tiempo tendríamos el mismo problema.

El segundo grupo trabajó en la creación de un nuevo protocolo que permitiera mayor capacidad de direccionamiento y sea adecuado para nuevos servicios sobre la Internet.

La primera propuesta que surgió fue la de reemplazar IP por el protocolo CLNP (Connection Less Network Protocol) de OSI (Open System Interconnection), a esta propuesta se la llamó TUBA, ya que utilizaba TCP (Protocolo de control de transmisión) y UDP (Protocolo para datagramas de usuario) sobre grandes direcciones, pero esto fue rechazado por numerosos miembros indicando que éste había sido copiado diez años antes de IPv4 introduciendo algunas modificaciones.

En 1992 Roberto Ullman propuso un nuevo protocolo IP llamado IPv7. La propuesta fue reelaborada en 1993 y asumió el nombre de TP/IX para indicar el cambio de dos protocolos el IP y el TCP al mismo tiempo. La propuesta tuvo ideas interesantes como la velocidad de los paquetes de procesamiento y un nuevo protocolo de ruteo llamado RAP. En 1994 la propuesta tuvo otra evolución tratando de definir un único formato para los paquetes CLNP e IPX (Protocolo similar a IP desarrollado para Novell) y asumiendo un nuevo nombre de CATNIP (Common Architecture for the Internet), la cual utilizaba varios protocolos de transporte como OSI/TP4, TCP y UDP.

Otra propuesta hecha en 1992 fue IP sobre IP, diseñada para utilizar dos capas IPv4 para limitar la escasez de direcciones en la Internet, una capa como backbone principal y una segunda capa dentro de áreas limitadas. En 1993 la

propuesta fue desarrollada y se llamó IPAE (IP Address Encapsulation) y fue válida como una solución de transición hacia SIP.

El SIP (Simple IP) fue propuesto por Steve Deering en Noviembre de 1992. Se basa en la idea de traer direcciones IP de 64 bits y eliminar algunos detalles obsoletos de IPv4. Esta propuesta fue inmediatamente aceptada por varias compañías quienes apreciaron su simplicidad.

Paúl Francis propuso el PIP (Paul´s Internet Protocol) la cual introducía significativas innovaciones en el ruteo haciéndolo más eficiente. En septiembre de 1993 PIP se combinó con SIP originando SIPP (Simple IP Plus).

SIPP intenta combinar la simplicidad de SIP y la flexibilidad de ruteo de PIP. Se ha diseñado para trabajar eficientemente en redes de alto performance, como ATM, pero también en redes de bajo performance como redes inalámbricas. SIPP tiene una pequeña cabecera y direcciones de 64 bits. Con SIPP la cabecera puede ser elaborada de manera eficiente por los routers y se puede extender para insertar nuevas opciones en el futuro.

La decisión fue tomada en Julio de 1994 y se adoptó a SIPP como base para IPv6 con la modificación en la longitud de bits, en vez de 64 se puso en 128 bits.

2.4 REQUISITOS QUE DEBE CUMPLIR IPv6[47]

2.4.1 ESPACIO DE DIRECCIONAMIENTO

Se debe tener una dirección IPv6 para cada usuario potencial de Internet. Se puede estimar que el crecimiento de la población en el mundo será de 10 billones de personas y se asume que cada persona tendrá más de un computador porque en el futuro los electrodomésticos, los dispositivos médicos y los aparatos en general serán computadoras.

[47] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997

Hoy en día están disponibles sistemas de iluminación domésticos donde las lámparas tienen una dirección y éstas son prendidas o apagadas por mensajes enviados por switches a través de un bus. En el Internet futuro el usuario deseará ordenar desde fuera de su casa que el horno empiece a cocinar el pavo, o recibir un mensaje de alarma detectando a un posible intruso y todo controlado desde su browser Internet usando una video cámara controlada remotamente. Los ejemplos son infinitos, y un estimado de 256 direcciones IPv6 para cada habitante no es poco realista.

Además del número de direcciones que se asignarán, es también importante considerar la eficiencia del esquema de asignación. "Un estudio más exacto realizado por Christian Huitema propone definir la eficiencia de asignación de direcciones H como la relación entre el logaritmo en base 10 del número de direcciones usadas y el número de bits de la dirección.

$$H = \frac{log\ (número\ de\ direcciones)}{número\ de\ bits}$$

En un esquema con una tasa de máxima eficiencia todas las direcciones son usadas y por tanto H es igual al logaritmo en base 10 de 2, es decir H = 0.301. Un análisis real de los esquemas de direccionamiento muestra que H varia entre 0.22 y 0.26."[48]

Christian Huitema realizó un análisis más profundo de algunas redes y concluyó que en el peor de los casos la eficiencia de asignación más pesimista era de 0.14. La decisión final es la de prever un millón de billón de hosts (10^{15}) que con H igual a 0.14 que resulta el peor caso, requiere direcciones de 107 bits. Por razones de implementación este debe ser un múltiplo de 32 bits por lo que se optó por tener direcciones IPv6 de 128 bits, es decir 16 octetos, con lo que se tienen 4 palabras de 32 bits.

[48] Tomado de Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997

2.4.2 DIRECCIONES MULTICAST Y ANYCAST[49]

Además de las direcciones unicast, IPv4 utiliza también el Multicast o direcciones de clase D para aplicaciones tales como Video Conferencia en el Internet. Estas direcciones Multicast se prevén en IPv6.

IPv6 introduce un nuevo tipo de direcciones llamadas "anycast". También son direcciones de grupo donde el más cercano es el único en responder a la fuente. El uso de direcciones anycast es muy interesante, como por ejemplo una dirección anycast al router más cercano o al servidor más cercano.

2.4.3 UNIFICAR INTRANET Y LA INTERNET[50]

IPv6 debe proveer un esquema de direccionamiento que permita unificar la Internet y las Intranets (redes locales de uso interno) superando las soluciones temporales de IPv4. Para este propósito, además de direcciones globales, también se han previsto direcciones locales y direcciones de enlace o conexión. Las direcciones locales se deben utilizar para nodos de red dentro de Intranets, mientras que las direcciones de conexión son usadas para numerar enlaces finales entre routers y por lo tanto son importantes solamente para ruteo y propósitos de administración.

2.4.4 USANDO MEJORES LANS

Cuando IPv4 opera sobre una LAN, frecuentemente necesita determinar la correspondencia entre una dirección IPv4 y una MAC address (dirección de control de acceso al medio, que son las direcciones físicas que se graban en el proceso de fabricación con el objetivo de que no existan dos direcciones iguales) y viceversa. IPv4 realiza esta función a través de un protocolo auxiliar llamado ARP (Address Resolution Protocol) que utiliza transmisiones broadcast de la capa MAC. Un paquete broadcast es un paquete que es recibido por todas las

[49] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997
[50] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997

estaciones y que causa una interrupción en todas las máquinas, incluidas aquellas que no utilizan el protocolo IP. Esta es una ineficiencia que debe ser corregida en IPv6 usando un método "neighbor discovery" (es un proceso que usa mensajes y solicita a un nodo direcciones multicast a fin de determinar la dirección de capa enlace) sobre una LAN, más eficiente que ARP utilizando Multicast y no transmisiones de broadcast.

2.4.5 SEGURIDAD[51]

La seguridad en IPv4 es manejada hoy en día a través de routers particulares o computadoras que hacen el papel de firewalls (es un software o hardware que pretenden dar seguridad a redes corporativas contra accesos no autorizados). Estos no pueden resolver problemas intrínsecos de la seguridad de IPv4, pero pueden contrarrestar muchas debilidades de los sistemas operativos y la administración superficial de seguridad que frecuentemente existe a nivel del computador.

IPv6 no ha sido desarrollado necesariamente para mejorar el estado de seguridad, pero no lo hace peor. De hecho, el IETF definió una serie de procedimientos de encriptación y autentificación que estarán disponibles en el protocolo IPv6 desde el inicio. Estos procedimientos deberán ser implementados de una manera compatible con IPv4. IPv6 tiene una administración cuidadosa del Enrutamiento de fuente, es decir la posibilidad de determinar a nivel de la estación Fuente el camino que seguirá un paquete IP. Esta opción si está disponible en IPv4 pero no siempre implementada o activada, esto es frecuentemente explotada por los hackers para intentar desviar los firewalls.

Muchos administradores de red pensarán en la disponibilidad de un procedimiento estándar de seguridad como el principal estimulo para migrar a IPv6.

---

[51] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997

2.4.6 ENRUTAMIENTO

Un tema central es el diseño de un protocolo que permita enrutar paquetes en el futuro Internet. El enrutamiento de IPv4 es el punto de partida, podemos ver que las tablas de enrutamiento de los routers de Internet están a punto de explotar. En efecto, si el CIDR no es usado, cada red simple deberá ser anunciada por una entrada en la tabla de enrutamiento. La introducción del CIDR permite anunciar un bloque de redes con direcciones contiguas (por ejemplo, la 195.1.3.0, la 195.1.4.0, la 195.1.5.0 y la 195.1.6.0) como una única entrada especificando cuantos bits pueden ser considerados como significativos (195.1.3.0/22 que es cada red con los primeros 22 bits iguales a 195.1.3.0)

En cualquier caso, el CIDR puede hacer poco si éste no está conectado con la asignación de direcciones. Si las direcciones son asignadas a ISPs (Internet Service Provider) y por éstos a los usuarios, el CIDR trabaja correctamente, desde un punto de vista teórico, todas las direcciones de un simple ISP se pueden anunciar por una única entrada. Se puede pensar en forma jerárquica de enrutamiento acompañada por una clase también jerárquica de asignación de direcciones limitada a la topología de la red.

Si se considera como la asignación de direcciones IPv4 se maneja hoy en día, una organización puede contactar a autoridades como INTERNIC (Internet Networks Information Center en América), APNIC (Asia Pacificic Network Information Centre en Asia y el Pacífico) y RIPE-NCC (RIPE Network Cordination Centre en Europa) para obtener direcciones que la organización usará independientemente de los ISP las cuales le permitirán conectarse. De esta manera la organización puede cambiar de ISP sin cambiar las direcciones. Con IPv6 si la organización cambia de ISP necesariamente tendrá que cambiar de direcciones.

El modelo de asignación de direcciones basado en la topología de red es aceptable en IPv6 solamente si los mecanismos de autoconfiguración (plug and

play) están disponibles, por ejemplo redes que asignan dinámicamente las direcciones a las estaciones.

IPv6 también preverá la posibilidad de tener una política de ruteo y QoS (Quality of Service). Un ejemplo de enrutamiento basado sobre una política en particular es la que determina la transmisión de paquetes hacia un destino dado por un camino determinado también por la dirección de la fuente.

Eventualmente será necesario que el ruteo IPv6 pueda proveer un buen soporte a la movilidad, por ejemplo a utilizadores que por medio de una computadora personal portable y de un teléfono portátil quieran conectarse ellos mismos al Internet en lugares continuamente diversos.

2.4.7 UN BUEN SOPORTE A ATM[52]

El gran esfuerzo industrial relacionado con el desarrollo de ATM (Asynchronous Transfer Mode) hará a esta tecnología uno de los más importantes actores en redes de área amplia y redes de área local. Los diseñadores de IPv6 han tratado de mejorar el soporte a ATM en IPv6. ATM es una red NBMA (Non-Broadband Multiple Access) y ésta garantiza el QoS.

Una red NBMA es una red de acceso multipunto que no provee un mecanismo simple para transmitir paquetes al resto de estaciones. IPv4 se ha diseñado para trabajar solo en canales punto a punto que tienen solamente dos puntos finales o sobre redes locales que tienen acceso múltiple, pero donde un paquete de transmisión a una estación o a todas las estaciones tiene exactamente el mismo costo. Otras redes NBMA son por ejemplo, X.25 y Frame Relay[53], pero la necesidad de dar un buen soporte IP a redes NBMA surgió solamente con ATM, debido al papel que esta tecnología jugará en el futuro.

---

[52] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997

[53] Frame Relay es una tecnología eficiente de conmutación de paquetes que permite la entrega confiable de paquetes sobre circuito virtuales

Para garantizar el QoS debe asociarse a cada dato de flujo un conjunto de requisitos de calidad. Por ejemplo, si el flujo de datos ha sido generado por una transferencia de archivos es muy importante que la tasa de pérdida sea igual a cero, mientras que el retardo al cual los paquetes están sujetos a lo largo del camino es irrelevante. Si el flujo de datos es generado por una fuente de audio o video, una cierta tasa de pérdida de datos puede ser tolerada, pero es fundamental garantizar un límite y menos variabilidad de retardos de un paquete a otro.

El QoS se puede utilizar solamente si es solicitado por la aplicación, una cosa que las aplicaciones de hoy no lo hacen.

2.4.8 EL CONCEPTO DE FLUJO[54]

Para simplificar la implementación de IPv6 sobre ATM y la administración del QoS, es necesario introducir el concepto de flujo. Un flujo es una secuencia de paquetes de alguna manera correlacionada y que debe ser tratada coherentemente por la capa IP. Los paquetes pertenecen al mismo flujo en base de parámetros como la dirección de fuente, la dirección de destino, el QoS, la autentificación y la seguridad.

No hay relación entre el concepto de flujo y otros conceptos como la conexión TCP: por ejemplo un flujo puede contener varias conexiones TCP. Por otra parte es importante enfatizar que la introducción del concepto de flujo ocurre sobre un protocolo que es sin conexión (frecuentemente llamado datagrama) y que por lo tanto el flujo no tiene el mismo propósito de los protocolos orientados a conexión, por ejemplo corrección de errores. Más en general, un flujo puede tener como destinos a una sola estación o un grupo de estaciones y por lo tanto se puede tener flujos unicast o multicast.

---

[54] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997

Una vez conocido el concepto de flujo se puede introducir el concepto de etiqueta de flujo con el cual marcamos los paquetes o datagramas reservando un campo especial en la cabecera IPv6. De esta manera IPv6 tiene la posibilidad, en el momento que recibe el paquete, de saber a que flujo pertenece examinando la etiqueta de flujo y de ésta manera conocer las necesidades del paquete en términos de QoS.

2.4.9 PRIORIDADES[55]

Si una aplicación no solicita un QoS, es de todos modos posible distinguir el tráfico generado por aplicaciones principales en función de sus requerimientos en tiempo real. Con este propósito un campo de "prioridad" de 4 bits se ha introducido en la cabecera IPv6 para distinguir 16 prioridades potenciales de tráfico. Hasta ahora las prioridades se han definido para email[56], ftp[57], nfs[58], telnet[59], enrutamiento y protocolos SNMP[60](Simple Network Management Protocol).

2.4.10 PLUG AND PLAY[61]

El protocolo DHCP (Dynamic Host Configuration Protocol) disponible en algunas implementaciones IPv4 se ha considerado como un buen punto de partida. La idea es desarrollar un protocolo DHCPv6 que permita la configuración automática de hosts y subredes, de manera que aprenda el valor por defecto que tienen los routers y a través de una interacción con el DNS (Domain Name Service) también una configuración automática de los nombres de los hosts.

---

[55] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997
[56]. Sistema de mensajería informática similar en muchos aspectos al correo ordinario pero más rápido
[57] Protocolo de Transferencia de Archivos utilizado para la transferencia de archivos entre diferentes máquinas a través de la red.
[58] Sistema de Archivos de Red
[59] Telnet es el protocolo estándar de Internet para realizar un servicio de conexión desde un terminal remoto
[60] Es un standard definido por la IETF para el manejo de la gestión de información
[61] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997

La implementación del DHCPv6 en todos los host IPv6 permitirá que los administradores de red reconfiguren direcciones para operar sobre el servidor primario DHCPv6.

2.4.11 MOVILIDAD[62]

La mayoría de usuarios de Internet no trabaja en su oficina, sino durante viajes. El usuario móvil está equipado usualmente con una computadora portable con una tarjeta de red, ésta se conecta a su teléfono celular o a una red pública vía radio.

IPv4 no provee ninguna ayuda a la movilidad. En efecto cada computadora tiene una dirección fija que pertenece a una red. Si el computador es conectado a una red diferente, los paquetes enviados a ella continúan alcanzando la red original y allí se pierden.

La solución está en asignar al usuario móvil dos direcciones, la primera "permanente" a su red en su organización y la segunda "dinámica" dependiendo del punto al cual está conectado en un instante dado del tiempo. El firewall de la organización, cuando está viajando el usuario, actúa como "proxy" para la dirección permanente y coloca un túnel seguro hacia el direccionamiento dinámico.

2.4.12 TRANSICIÓN DE IPv4 A IPv6[63]

Una gran cantidad de usuarios considerarán la transición a IPv6 como algo a lo cual es necesario resignarse para obtener ventajas potenciales. Cambiar el software de la red es muy similar a cambiar la versión del sistema operativo, es un paso que potencialmente trae algunas incompatibilidades y causa la necesidad de poner al día el hardware y el software.

---

[62] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997
[63] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997

El IETF decidió diseñar una migración estratégica basada sobre un acercamiento "dual-stack" (sistema de doble pila IP desarrollado para la transición de IPv4 a IPv6), pero esto será un campo donde los vendedores de computadores y redes lucharán fuertemente para simplificar la vida del usuario y ganar mercado. De hecho muy pocos usuarios tendrán la posibilidad para migrar en un momento dado, la mayor parte de organizaciones tendrá meses o años durante los cuales IPv6 deberá coexistir con IPv4.

Por esta razón IETF decidió que IPv4 e IPv6 deberán tener dos diferentes protocolos con dos correspondientes y separados pilas de protocolos. Cuando una estación recibe una trama de su red local, el Tipo de Protocolo permite distinguir si la trama contiene un paquete IPv4 o IPv6.

Por lo tanto el primer campo de los paquetes IPv4 o IPv6, representa la versión del protocolo, el cual deberá ser utilizado como pila IPv4 cuando se reciban solamente paquetes IPv4 o como pila IPv6 cuando se reciban paquetes IPv6.

Uno de los pasos críticos en la transición deberá ser la administración paralela de las direcciones IPv4 e IPv6. Será necesario una puesta al día de los servidores de DNS y DHCP. Una estación de pila dual deberá usar direcciones IPv4 (32 bits) para comunicarse con otra estación IPv4, mientras que usará direcciones IPv6 (128 bits) para comunicarse con otras estaciones IPv6.

Para este acercamiento, las islas IPv6 deberán estar interconectadas. Esta conexión será implementada a través de una serie de túneles en Internet y por lo tanto en IPv4, ésta formará una red llamada 6-bone[64]. Este acercamiento se basa en la experiencia positiva del Mbone[65], la red usada para la video conferencia en el Internet que se ha implementado con éxito siguiendo la misma filosofía.

---

[64] Denominado así al Backbone que utiliza el protocolo IPv6

[65] Red de banda ancha y alta velocidad que permite actualmente la realización de audio y videoconferencias entre centenares de usuarios remotos a través de varios canales de vídeo y de audio

El 6-bone crecerá y algunas islas se interconectarán directamente usando IPv6, sin la necesidad de túneles. Un incremento de número de máquinas se comunicará utilizando IPv6 y entonces el día de IPv4 llegará, cuando todas las computadoras que corran solamente bajo el protocolo IPv4 pierdan su conectividad global y directa al Internet.

2.5 CABECERA Y OPCIONES DE IPv6[66]

En IPv6 la cabecera ha sido simplificada enormemente con respecto a IPv4. Muchos de los campos se han hecho opcionales o se han eliminado, la razón para esto ha sido disminuir al máximo el coste de procesamiento de los paquetes, debido al aumento considerable de las direcciones, cabe anotar que las direcciones IPv6 son cuatro veces más grandes que en IPv4, pero las cabeceras son solo dos veces más grandes que la cabecera IPv4.

Las cabeceras de opciones de IPv6 definidas son:

- Routing
- Fragmentación
- Hop-by-Hop
- Autenticación
- Seguridad de Encapsulación
- Opción End to End

2.6 FORMATO DE LA CABECERA IPv6[67,68]

La cabecera IPv6 contiene menos información que la cabecera del datagrama IPv4. En el datagrama IPv6 las opciones y algunos campos fijos que aparecen en el datagrama IPv4 se han reemplazado por cabeceras de extensión, es decir el

[66] http://a01-unix.gdyc.inf.uc3m.es/~baudoux/intro.htm
[67] http://www.everex.es/ip.htm
[68] RFC 1752 The recommendation for the IP Next Generation Protocol

cambio en los encabezados en los datagramas se refiere a cambios en el protocolo:

- La alineación es ahora en múltiplos de 64 bits en vez de 32 bits.
- El campo longitud de cabecera ha sido eliminado y el campo longitud de datagrama se ha reemplazado por el campo Longitud de Carga.
- El tamaño de los campos de dirección fuente y destino se ha incrementado a 16 octetos cada uno.
- Se ha movido de los campos fijos en la cabecera, la información de fragmentación hacia un encabezado de extensión.
- Se ha reemplazado el campo Tiempo de Vida por el de Límite de Saltos.
- El campo Tipo de Servicio es cambiado por el de Etiqueta de Flujo.

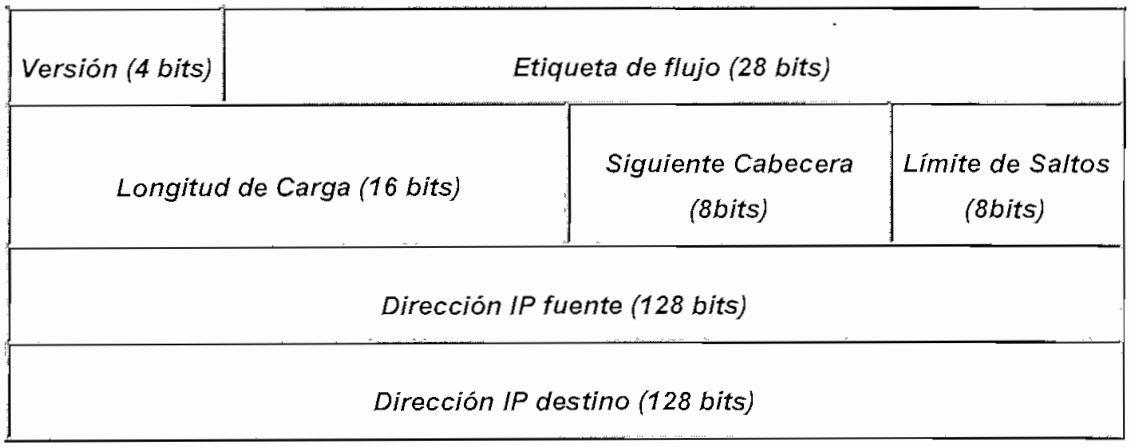


Fig. 2.2 Formato de la cabecera IPv6

2.6.1 VERSIÓN

Este campo ocupa 4 bits y especifica la versión del protocolo. En este caso es la versión 6.

2.6.2 ETIQUETA DE FLUJO[69]

El campo Etiqueta de flujo en la cabecera contiene información que los ruteadores utilizan para dar una prioridad y flujo específicos a un datagrama, este campo se subdivide en 2 subcampos:

Fig. 2.3 Subcampos de una Etiqueta de Flujo[70]

El campo **TCLASS** especifica la clase de tráfico para el datagrama, es decir nos da la prioridad que el remitente desea para los paquetes enviados, respecto a los demás paquetes enviados por él mismo. Los valores del 0 al 7 se emplean para especificar la sensibilidad al tiempo del tráfico controlado por flujo, es decir los paquetes para los cuales el remitente espera una respuesta en caso de congestión, por ejemplo tráfico TCP; los valores del 8 al 15 se utilizan para especificar una prioridad para tráfico que no es de flujo, paquetes que no deben ser respondidos en caso de congestión. El valor más bajo ocho, se usará cuando el remitente está dispuesto a que sus paquetes sean descartados en caso de congestión por ejemplo Video en Alta calidad. El valor más alto quince, cuando el remitente está muy poco dispuesto a que algún paquete sea descartado, por ejemplo Audio de baja calidad.

El campo de 24 bits, **Identificador de Flujo**, es usado para indicar que los paquetes sean tratados de forma especial por los routers, como en servicios de alta calidad o en tiempo real. El flujo se debe entender como un conjunto de paquetes que requieren un trato especial.

Todos los paquetes del mismo flujo deben tener valores similares en los campos de dirección origen, dirección destino, TCLASS y Etiqueta de flujo.

---

[69] Redes Globales de Información con Internet y TCP/IP. Douglas E. Comer, Tercera edición 1996.

[70] Redes Globales de Información con Internet y TCP/IP. Douglas E. Comer, Tercera edición 1996

2.6.3 LONGITUD DE LA CARGA[71]

Especifica el número de octetos transportados en un datagrama, excluyendo a la cabecera, así un datagrama IPv6 puede contener 64K octetos de datos. Si el valor es cero, indica que el tamaño de la carga vendrá especificado como Carga Jumbo, en una opción salto a salto.

2.6.4 SIGUIENTE CABECERA[72]

Este campo de 8 bits identifica el tipo de cabecera que sigue a la cabecera IPv6. Es coherente con los valores de campo protocolo en IPv4.

2.6.5 LIMITE DE SALTOS (HOP LIMIT)[73]

Es correspondiente al campo TIME TO LIVE (tiempo de vida) de IPv4. A diferencia del IPv4, el cual interpreta un tiempo límite como una combinación de conteo de saltos y tiempo máximo, el IPv6 interpreta el valor como un límite estricto del máximo número de saltos que un datagrama puede realizar antes de ser desechado.

2.6.6 DIRECCIÓN ORIGEN

Este campo ocupa 128 bits y corresponde a la dirección de origen.

2.6.7 DIRECCIÓN DESTINO

Este campo ocupa 128 bits y corresponde a la dirección de destino.

---

[71] http://www.everex.es/ip.htm
[72] http://www.everex.es/ip.htm
[73] RFC 1752 The Recommendation for the Next Generation Protocol

2.7 CABECERAS EXTENDIDAS[74,75]

En IPv6 cierta información es codificada en cabeceras que deben colocarse en el paquete entre la cabecera IPv6 y la cabecera de la capa de transporte.

Existen varias cabeceras extendidas, cada una de ellas identificada por un valor en el campo *siguiente cabecera.* Un paquete IPv6 puede contener ninguna, una o más cabeceras extendidas.

Las cabeceras extendidas apenas son examinadas o manipuladas por los nodos alcanzados por el paquete a lo largo de su camino hasta que éste llega al nodo (o a cada grupo de nodos en el caso del Multicast) identificados por el campo de dirección de destino de la cabecera IPv6. En este momento se trata la primera cabecera extendida, o la cabecera de transporte en el caso de ausencia de cabeceras extendidas. El contenido de cada cabecera determinará si es necesario tratar la cabecera siguiente.

La única excepción es la cabecera de la opción salto a salto, que lleva información que deberá ser examinada por los nodos de la red. Esta cabecera "Hop by Hop" o salto a salto, cuando está presente, tiene que seguir inmediatamente a la cabecera IPv6.

Cada cabecera extendida es de una longitud de un múltiplo de 8 bytes, para conservar una alineación a 8 bytes en las cabeceras extendidas posteriores.

Si en el procesamiento de las cabeceras, un nodo se encuentra con un valor en *siguiente cabecera* que le es desconocido, el paquete debe ser descartado, y un nivel superior (ICMP Protocolo de Mensajes de Control Internet), se encargará de enviar un error al origen.

---

[74] http://a01-unix.gsyc.inf.uc3m.es/~baudoux/intro.htm
[75] RFC 1752 The Recommendation for the IP Next Generation Protocol

Cuando hay más de una cabecera extendida en un mismo paquete, las cabeceras deben aparecer en el orden siguiente:

- Cabecera IPv6
- Cabecera de opciones Hop by Hop
- Cabecera de Enrutamiento
- Cabecera de Fragmentación
- Cabecera de Autenticación
- Cabecera de Extremo a Extremo

Cada tipo de cabecera debe aparecer una sola vez en el paquete con la excepción de las opciones de destino, que pueden aparecer dos veces, en el orden indicado anteriormente (también en el caso de una encapsulación IPv6 en IPv6, donde cada cabecera IPv6 encapsulada debe ser seguida por su propia cabecera extendida).

Los nodos que soporten IPv6, deben aceptar y procesar las cabeceras en cualquier orden que aparezcan y también si aparecen dos o más veces, a excepción de las opciones salto a salto, que deben aparecer inmediatamente después de la cabecera IPv6. Se recomienda que los nodos que envíen paquetes IPv6 sigan el orden recomendado.

2.7.1 OPCIONES TLV (Type-Length-Value)

Dos de las cabeceras definidas (salto a salto y opciones de destino), llevan un número variable de opciones, las cuales a su vez tiene longitud variable TLV. Estas opciones tienen la siguiente estructura:

- **Tipo de opción:** Este campo ocupa 1 octeto, y actúa como identificador de cada opción específica.

Tipo de Opción (8 bits)	Longitud datos opción (8bits)	Datos (Long Variable)

Fig. 2.4 Estructura de las opciones TLV[76]

- **Longitud de datos:** este campo ocupa 1 octeto, e indica la longitud del campo de datos, medida en octetos.

- **Datos:** Este campo tiene una longitud variable, en el se encuentran los datos específicos de cada opción.

La secuencia de opciones debe ser procesada en el orden en que aparezcan, el receptor no puede examinar la cabecera en busca de una opción y procesarla antes que las anteriores.

El campo *Tipo de Opción* está codificado de tal manera que los dos bits de mayor peso especifican las acciones a tomar en caso que el nodo no reconozca la opción:

00 Descartar la opción y seguir procesando el paquete
01 Descartar el paquete entero
10 Descartar el paquete y enviar un mensaje de error ICMP al origen
11 Descartar el paquete y enviar un mensaje de error ICMP al origen, si y solo si el paquete no tiene una dirección destino multicast.

El tercer bit de mayor peso especifica si los datos específicos de la opción pueden cambiar durante el recorrido del paquete. Esto es útil cuando existe una cabecera de autenticación. Cualquier mecanismo de autenticación deberá tomar como 0´s los datos que puedan cambiar en ruta. Los valores son:

0 Datos de la opción NO pueden cambiar en ruta.
1 Datos de la opción pueden cambiar en ruta.

[76] http://www.everex.es/ip.htm

Las distintas opciones pueden tener diferentes requerimientos de alineamiento, estos requerimientos se especifican en la forma xn+y. Por ejemplo, 2n significa que la opción debe encontrarse desplazada del comienzo de la cabecera en octetos múltiplos de 2.

Para mantener el alineamiento, existen dos opciones de relleno. Si sólo se requiere un octeto, éste se coloca todo a 0´s sin más. Si se necesita más de un octeto, se usa la siguiente estructura:

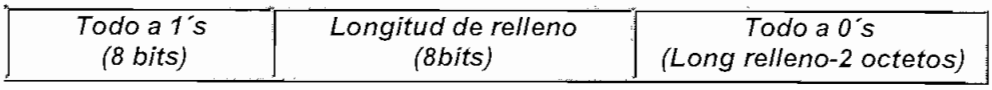


Fig. 2.5 Estructura de relleno para más de un octeto[77]

La longitud se especifica en octetos, sin tener en cuenta el preámbulo y el campo de longitud, por tanto, el número de octetos en el relleno propiamente dicho es longitud-2.

2.7.2 CABECERA DE OPCIONES SALTO A SALTO (NODO POR NODO)[78]

La cabecera extendida de Opciones Salto a Salto contiene información que deberá ser examinada por cada nodo que encamine el paquete hacia su destino. Este tipo de cabecera se identifica con el valor 0 en el campo *siguiente cabecera*.

Su formato es el siguiente:

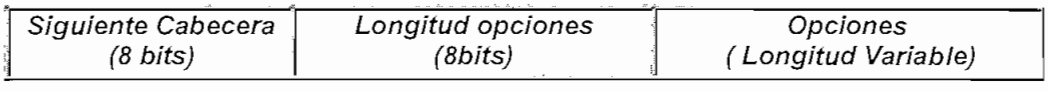


Fig. 2.6 Formato de la Cabecera Opciones Salto a Salto[79]

[77] http://www.everex.es/ip.htm

[78] RFC 1752 The Recommendation for the IP Next Generation Protocol

[79] http://www.everex.es/ip.htm

- **Siguiente Cabecera:** Ocupa 1 octeto e identifica el tipo de cabecera existente inmediatamente después de las Opciones Salto a Salto. Sus valores son idénticos al campo Protocol de la versión IPv4.

- **Longitud opciones:** Este campo ocupa 1 octeto e indica la longitud de la cabecera en octetos, sin incluir los ocho primeros.

- **Opciones:** Es de longitud variable y contiene opciones codificadas en TLV (Type Lenght Value).

Además de las opciones con la estructura descrita, existe una opción especial, la Carga Jumbo con la siguiente estructura:

194 (Identificador) *8 bits*	*4 (Long. Opciones)* *8 bits*	*Longitud Carga Jumbo* *32 bits*

Fig. 2.7 Estructura de la opción Carga Jumbo[80]

La opción Carga Jumbo, es utilizada para enviar paquetes con cargas superiores a los 65535 octetos. La longitud especificada por la Carga Jumbo es el tamaño total del paquete, excluyendo la cabecera IPv6 e incluyendo la cabecera de opciones Salto a Salto.

La longitud determinada debe ser siempre superior a 65535, si se recibe un paquete con una Carga Jumbo que indique un tamaño de paquete igual o menor a 65535, ICMP se encargará de enviar un error.

Cada paquete cuya longitud esté especificada por una opción Carga Jumbo, debe tener a 0 el campo *longitud de la carga* en la cabecera IPv6, además, la opción Carga Jumbo no puede ser usada en un paquete conteniendo un fragmento. El incumplimiento de cualquiera de estas restricciones provocará un error ICMP.

[80] RFC 1752 The Recommendation for the IP Next Generation Protocol

2.7.3 CABECERA DE ENRUTAMIENTO[81]

La cabecera de enrutamiento es utilizada por un emisor para establecer una lista de nodos intermedios que debe seguir el paquete para llegar a su destino. Esta forma particular de cabecera de enrutamiento está diseñada para soportar el "protocolo de enrutamiento a petición del emisor" (Source Demand Routing Protocol, SDRP).

Su formato es el siguiente:

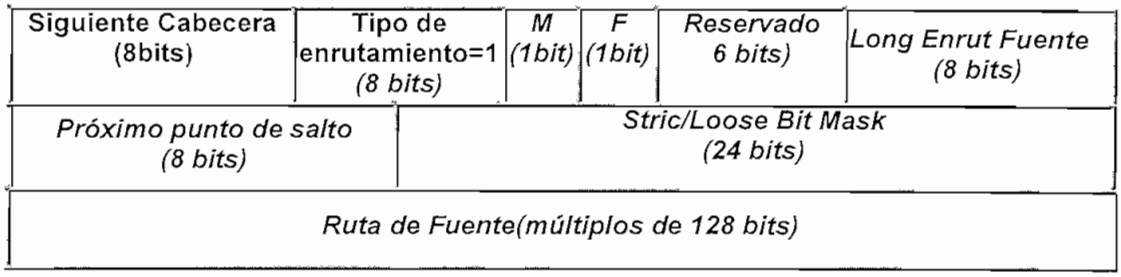


Fig. 2.8 Estructura de la Cabecera de Enrutamiento[82]

- **Siguiente cabecera:** Este campo ocupa 1 octeto e identifica el tipo de cabecera que sigue inmediatamente a la cabecera de encaminamiento. Sus valores son idénticos a los del campo Protocol de la versión IPv4.

- **Tipo de enrutamiento:** Indica el tipo de enrutamiento soportado por esta cabecera. Su valor es 1.

- **Bit MRE (Must Report Errors):** Si este bit está a 1 y un router no puede emitir correctamente la lista de enrutamiento de fuente, el router genera un mensaje de error ICMP. En el caso que este bit esté a 0, el router no genera este mensaje.

[81] http://a01-unix.gsyc.inf.uc3m.es/~baudoux/intro.htm
[82] http://a01-unix.gsyc.inf.uc3m.es/~baudoux/intro.htm

- **Bit F (failure of Source Route Behavior):** Si este bit está a 1, esto indica que un router no puede enrutar más lejos un paquete, como se especifica en la ruta de fuente.

- **Reservado:** Ocupa 6 bits, inicializado a 0 por el emisor, ignorado por el receptor.

- **Longitud de Enrutamiento Fuente (LEF):** De 8 bits, es el número de elementos o nodos que hay en una cabecera de encaminamiento SDRP. Este campo no debe exceder el valor de 24.

- **Próximo Punto de Salto:** Apunta a los elementos o nodos que hay que alcanzar. Es inicializado a 0 para alcanzar al primer nodo del enrutamiento fuente. Cuando es igual al campo Longitud de ruta Fuente, significa que el enrutamiento de fuente está terminado.

- **Strict/Loose Bit Mask:** Este campo ocupa 24 bits, esta máscara se utiliza para que un nodo opte por un camino. Si el valor del Próximo punto de salto es N, significa que el N-ésimo bit del Strict/Loose Bit Mask está a 1, esto indica que el siguiente nodo es un nodo vecino del anterior; 0 significa que no es necesariamente un nodo vecino.

- **Enrutamiento Fuente:** Es un múltiplo de 128 bits, y constituye una lista de direcciones IPv6 que indica el camino que debe seguir el paquete, se numeran de 1 a n y pueden aparecer como máximo 23. Esta puede contener un conjunto de direcciones de tipo unicast.

2.7.4 CABECERA DE FRAGMENTACIÓN[83]

En IPv6 la fragmentación está restringida a la fuente original. Antes de enviar tráfico de información, una fuente debe realizar una técnica de *Path MTU*

[83] http://a01-unix.gsyc.inf.uc3m.es/~baudoux/intro.htm

Discovery (descubrir la MTU de la ruta) para identificar la MTU (Maximum Transfer Unit) mínima a lo largo de la trayectoria hasta el destino. Antes de enviar un datagrama, la fuente fragmenta el datagrama de manera que cada fragmento sea menor que el Path MTU. Así, la fragmentación es de extremo a extremo; no son necesarias fragmentaciones adicionales en ruteadores intermedios. La cabecera de fragmentación se distingue por un valor del campo Siguiente Cabecera igual a 44.

La cabecera tiene el siguiente formato.

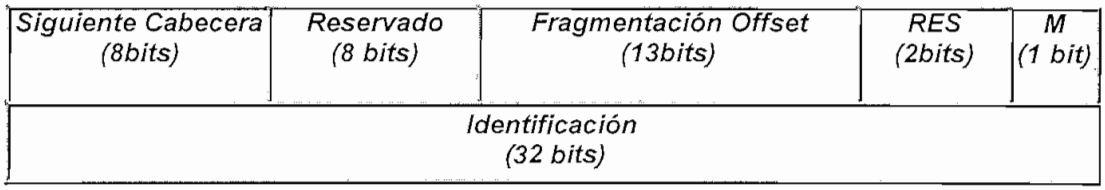


Fig. 2.9 Formato de la cabecera fragmentación[84]

- **Siguiente cabecera:** Ocupa 8 bits e identifica el tipo de la cabecera que sigue inmediatamente a la cabecera de fragmentación.

- **Reservado:** Este campo ocupa 8 bits. El origen lo pone a 0 y es ignorado por el receptor.

- **Fragmentación Offset:** Este campo ocupa 13 bits e indica en qué parte del datagrama actual va este fragmento. El primer fragmento estará en el lugar número 0. El valor de este campo es un múltiplo de 8 octetos.

- **RES:** De 2 bits inicializado a 0 por el emisor e ignorado por el receptor.

- **Flag M:** Si este bit está 1, significa que queda uno o más fragmentos, si es 0, indica que es el último fragmento.

[84] http://a01-unix.gsyc.inf.uc3m.es/~baudoux/intro.htm

- **Identificación:** Ocupa 32 bits y es un valor asignado al paquete de origen que es diferente de los demás paquetes fragmentados recientemente con la misma dirección fuente, la misma dirección destino y el mismo valor del campo siguiente cabecera. Este campo permite identificar el datagrama para asegurar el reensamblaje de los paquetes. El número de identificación está contenido en la cabecera de todos los fragmentos.

2.7.5 CABECERA DE AUTENTICACIÓN[85]

La cabecera de autenticación es utilizada para autenticar y asegurar la integridad de los paquetes IPv6. El no rechazo de los paquetes se obtiene con un algoritmo de autenticación usado con la cabecera de autenticación, ésta cabecera viene determinada por el valor 51 del campo Siguiente Cabecera, y tiene el siguiente formato:

Siguiente Cabecera (8 bits)	*Auth Data Lenght (8bits)*	*.Reservado (16 bits)*
Asociación de Seguridad ID (32 bits)		
Datos de Autenticación (Variable)		

Fig. 2.10 Formato de la cabecera de Autenticación[86]

- **Siguiente Cabecera:** Ocupa 8 bits e identifica el tipo de cabecera que sigue inmediatamente a la cabecera de autenticación.

- **Authentication Data Lenght:** Campo de 8 bits. Es la longitud del campo Datos de Autenticación, múltiplo de 8 octetos.

- **Reservado:** Campo de 16 bits, inicializado a 0 al principio de la emisión e ignorado en la recepción.

[85] http://www.everex.es/ip.htm
[86] http://www.everex.es/ip.htm

- **Asociación de Seguridad ID:** Ocupa 32 bits, combinado con la dirección fuente, indica al o los destinatarios el tipo de seguridad al cual el paquete está sometido.

- **Datos de Autenticación:** De longitud variable y múltiplo de 8 octetos. El algoritmo específico necesario para autenticar el origen del paquete y para asegurar su integridad con respecto al tipo de seguridad asociado.

2.7.6 CABECERA DE CONFIDENCIALIDAD[87,88]

Intenta dar confidencialidad e integridad encriptando los datos a proteger y colocándolos en la parte de la cabecera de confidencialidad. Dependiendo de las exigencias de seguridad del usuario, se podrá encriptar la trama del nivel de transporte (UDP o TCP) o el datagrama entero. Este enfoque con encapsulación es necesario para asegurar una confidencialidad completa del datagrama original.

La cabecera de confidencialidad funciona entre hosts, entre hosts y un gateway de seguridad o entre dos gateway de seguridad. Esto permite asegurar una red sin costes financieros elevados que transfieren un tráfico seguro sobre partes de la red que no lo son.

Su formato es el siguiente:

- **Identificador de Asociación de Seguridad (SAID):** Campo de 32 bits, identifica el tipo de seguridad del datagrama. Si no se ha establecido ninguna asociación de seguridad, el valor de este campo será 0x0000. Una asociación de seguridad es unilateral. Una comunicación confidencial entre dos estaciones debe tener normalmente dos SAID (uno para cada uno de los destinos). La estación de destino utiliza la combinación del valor del SAID y de la dirección fuente para distinguir la asociación correcta.

---

[87] http://www.everex.es/ip.htm
[88] RFC 1752 The Recommendation for the IP Next Generation Protocol

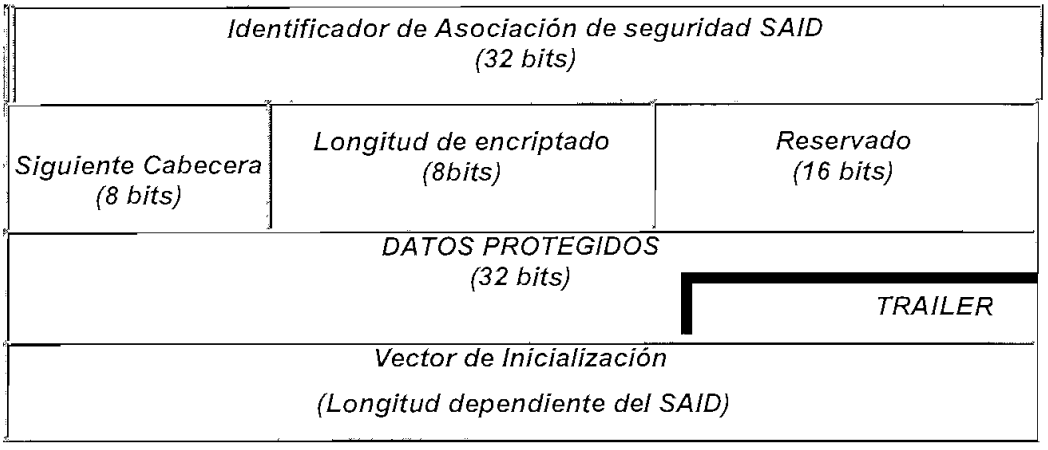


Fig. 2.11 Formato de la cabecera de Confidencialidad[89]

- **Vector de Inicialización (longitud dependiente del SAID):** Este campo es opcional y su valor depende del SAID utilizado. Por ejemplo, el campo puede contener datos de sincronización de criptografía para un algoritmo de codificación. Puede contener también un vector de inicialización criptográfica.

  La implantación de una cabecera de confidencialidad utiliza un valor de SAID para determinar si el campo no está vacío, y si es el caso, evalúa la longitud del campo y lo utiliza.

- **Siguiente cabecera:** Campo de 8 bits, identifica el tipo de cabecera que sigue inmediatamente a la cabecera de confidencialidad.

- **Reservado:** De 16 bits es ignorado por el receptor.

- **Longitud:** Campo de 8 bits que indica la longitud de la cabecera codificada (es un múltiplo de 8 octetos), a excepción de los 8 primeros octetos.

[89] http://www.everex.es/ip.htm

- **Datos Protegidos, encriptados:** De longitud variable. Este campo puede contener encapsulado un datagrama IPv6 completo, una secuencia de opciones IPv6, y por último, el paquete del nivel de transporte.

- **Trailer:** este campo es utilizado para hacer de sobre relleno (necesario en algunos algoritmos) o para registrar datos de autenticación utilizados en un algoritmo de criptografía que proporcione confidencialidad sin autenticación. Este campo está presente únicamente si el algoritmo lo necesita.

2.7.7 CABECERA DE EXTREMO A EXTREMO[90]

La cabecera de opciones extremo a extremo (end to end) da una información opcional que debe ser controlada por el o los nodos destino del paquete. Tiene el mismo formato que la cabecera de opción salto a salto.

2.8 DIRECCIONAMIENTO IPv6[91]

Las direcciones IPv6 tiene una longitud de 128 bits, y pueden identificar a nodos individuales o a conjuntos de nodos. Existen tres tipos de direcciones IPv6, denominadas:

- **Unicast (Unidifusión):** La dirección de destino especifica un solo host, el datagrama deberá rutearse hacia el destino a lo largo de la trayectoria más corta.

- **Anycast (Grupo):** El destino es un conjunto de hosts en el que todos comparten un solo prefijo de dirección (si están conectadas a la misma red física); el datagrama deberá rutearse hacia el grupo a través de la trayectoria más corta y después, entregarse exactamente a un host del grupo como por ejemplo el miembro más cercano.

---

[90] http://a01-unix.gsyc.inf.uc3m.es/~baudoux/intro.htm

[91] http://www.everex.es/ip.htm

- **Multicast (Multidifusión):** El destino es un conjunto de hosts, posiblemente en múltiples localidades. Una copia del datagrama deberá entregarse a cada miembro del grupo que emplee hardware de multidifusión o de difusión si están disponibles.

PREFIJO BINARIO	*TIPO DE DIRECCIÓN*	*PARTE DEL ESPACIO DE DIRECCION*
0000 0000	*Reservado (compatible con IPv4)*	*1/256*
0000 0001	*No asignado*	*1/256*
0000 001	*Direcciones NSAP*	*1/128*
0000 010	*Direcciones IPX*	*1/128*
0000 011	*No asignado*	*1/128*
0000 1	*No asignado*	*1/32*
0001	*No asignado*	*1/16*
001	*No asignado*	*1/8*
010	*Direcciones Unicast proveedor asignado*	*1/8*
011	*No asignado*	*1/8*
100	*Reservado (geográfico interconexión neutral)*	*1/8*
101	*No asignado*	*1/8*
110	*No asignado*	*1/8*
1110	*No asignado*	*1/16*
1111 0	*No asignado*	*1/32*
1111 10	*No asignado*	*1/64*
1111 110	*No asignado*	*1/128*
1111 1110 0	*No asignado*	*1/512*
1111 1110 10	*Para uso de enlaces locales*	*1/1024*
1111 1110 11	*Para uso de sitios locales*	*1/1024*
1111 1111	*Utilizado para multidifusión*	*1/256*

Fig. 2.12 Tipo específico de direcciones IPv6[92]

[92] http://a01-unix.gsyc.inf.uc3m.es/~baudoux/intro.htm

El tipo específico de direcciones IPv6 es indicado por los primeros bits de la dirección. El campo de longitud variable que incluye estos primeros bits es llamado *Format Prefix* (FP).

Esto soporta la asignación directa de proveedores de direcciones, direcciones de uso local y direcciones multicast. El espacio está reservado para direcciones NSAP (Punto de Acceso al servicio de Red) y direcciones IPX. El resto de espacio de direcciones está sin asignar para utilizaciones futuras. Estas pueden ser utilizadas para la expansión del uso existente (por ejemplo, proveedor adicional de direcciones, etc.) o nuevos usos.

El espacio de direcciones está dividido en NSAP, IPX, unicast basado en proveedor geográfico, direcciones de ámbito local y direcciones multicast. Esto es solo el 15% de todo el espacio de direcciones . El resto está reservado para usos futuros.

2.8.1 REPRESENTACIÓN DE DIRECCIONES[93]

Existen tres formas convencionales de representar las direcciones IPv6:

- 1.- La forma más aceptada es mediante la estructura **x:x:x:x:x:x:x:x**, donde las **x** representan los valores hexadecimales de los ocho bloques de dos octetos cada uno.

  Ejemplos:

  FEDC:BA98:7654:3210:FDCE:7564:BA98:7651

  1080:0:0:0:8:8000:200C:418A

[93] http://a01-unix.gsyc.inf.uc3m.es/~baudoux/intro.htm

Hay que destacar que no es necesario escribir todos los ceros que hay por delante de un nombre hexadecimal en un campo individual, pero se ha de tener por lo menos una cifra en cada campo.

- 2. - El método de alocación de direcciones IPv6 demuestra que es cómodo colocar bits a 0 en medio de las direcciones. Para una escritura fácil, una sintaxis adecuada sería suprimir estos ceros. La expresión de dos "::" indicaría uno o varios grupos de 16 bits iguales a 0. Por ejemplo:

  1080:0:0:0:8:800:200C:417C podría representarse como:
  1080::8:800:200C:417C

  FF01:0:0:0:0:0:0:43 podría representarse como FF01::43

  Sólo puede usarse "::" una vez en una dirección.

- 3. - Otra forma alternativa, a veces más cómoda cuando estamos en un entorno mixto de nodos IPv6 e IPv4, es **x:x:x:x:x:x:d.d.d.d**, donde los **x** son valores hexadecimales (6 grupos de 16 bits) y los **d** son valores decimales (4 grupos de 8 bits en la representación estándar de IPv4).

  Ejemplos:
  0:0.0:0:0:0:13.1.68.3 o ::13.1.68.3

  0:0:0:0.0:1:129.144.52.38 o ::1:129.144.52.38

2.8.2 DIRECCIONES UNICAST[94]

Una dirección unicast IPv6 tiene una estructura similar a una dirección IPv4 usando CIDR.

[94] http://www.everex.es/ip.htm

Existen múltiples formatos de dirección unicast, un nodo en Internet puede tener más o menos conocimiento de la estructura de las direcciones, dependiendo del papel que juegue en Internet. Como mínimo, un nodo considerará una dirección IPv6 como un identificador sin estructura interna.

Usando el valor de la máscara IP, pueden indicarse prefijos de red de longitud variable.

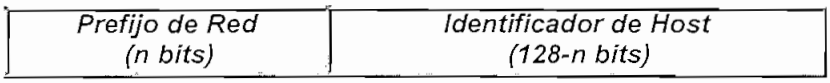


Fig. 2.13 Prefijo de Red de una dirección IPv6[95]

Los nodos pueden tener un conocimiento más profundo de la jerarquía de direcciones, dependiendo del papel que desempeñen en la jerarquía de enrutamiento.

Ejemplos de direcciones *unicast:*

- Direcciones MAC (IEEE 802) para redes locales.

  Siendo Identificador de host la dirección MAC del host. Para redes locales que no usen direcciones MAC, otros tipos de direcciones del nivel de enlace pueden ser usados.

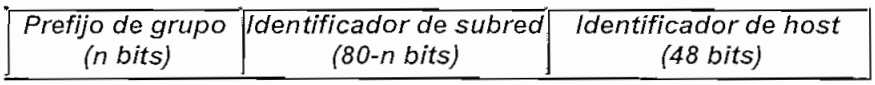


Fig. 2.14 Direcciones MAC para Redes Locales[96]

[95] http://www.everex.es/ip.htm

[96] http://www.everex.es/ip.htm

- Para sistemas que requieran por su tamaño más niveles de jerarquía, la dirección puede dividirse en múltiples niveles, por ejemplo:

Identificador grupo (g bits)	Identificador área (a bits)	Identificador subred (s bits)	Identificador host (128-g-a-s bits)

Fig. 2.15 Jerarquía de direcciones IPv6 por su tamaño[97]

- Para direcciones basadas en proveedor, tenemos la siguiente estructura:

010 (3 bits)	Id. Registro (n bits)	Id. Proveedor (m bits)	Id. suscriptor (s bits)	Id. Intra-suscriptor (125-n-m-s bits)

Fig. 2.16 Jerarquía de direcciones IPv6 basadas en proveedor[98]

Esta estructura refleja la jerarquía, un registro asigna las direcciones de un grupo de proveedores de servicios (por ejemplo backbones o redes regionales), que asignen direcciones a sus suscriptores.

2.8.3 DIRECCIONES ESPECIALES UNICAST[99]

- Dirección 0:0:0:0:0:0:0:0, esta dirección no puede ser asignada a ningún nodo ya que indica la ausencia de dirección. Puede usarse, como dirección origen al inicializar un host, antes que éstos conozcan su propia dirección IP. En ningún caso podrá aparecer como dirección destino.

- Dirección 0:0:0:0:0:0:0:1, ésta es la dirección del *bucle local*, puede ser usada por un host para enviarse un datagrama a él mismo. No podrá aparecer como dirección origen. Un datagrama enviado a la dirección de bucle local no saldrá

[97] http://www.everex.es/ip.htm
[98] http://www.everex.es/ip.htm
[99] http://www.everex.es/ip.htm

al medio, puede ser usada, por ejemplo, para comunicación entre los procesos de un nodo.

2.8.4 DIRECCIONES UNICAST IPv6 CONTENIENDO DIRECCIONES IPv4[100]

Existen dos formas de codificar direcciones IPv4 en direcciones IPv6. La primera se usa en nodos que puedan gestionar ambos protocolos, tanto IPv6 como IPv4. Las direcciones se codifican de la siguiente manera:

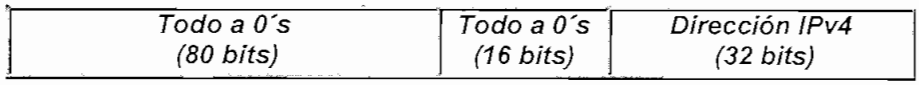


Fig. 2.17 Direcciones Unicast para nodos que puedan manejar los protocolos IPv4 e IPv6[101]

La segunda forma se usa para representar las direcciones de nodos que sólo soporten IPv4, antes de la conversión de IPv6 a IPv4, el datagrama llevará una dirección con la siguiente estructura:

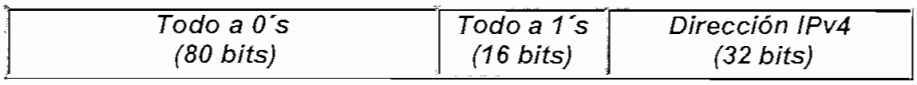


Fig. 2.18 Direcciones Unicast para nodos que soporten únicamente IPv4[102]

2.8.5 USO LOCAL DE DIRECCIONES UNICAST IPv6[103]

Existen dos tipos de direcciones IPv6 de uso local, estos tipos son el *enlace local* (Link-Local) y el *grupo local* (Site-Local).

La estructura de dirección ***enlace local*** es la siguiente:

[100] http://www.everex.es/ip.htm
[101] http://www.everex.es/ip.htm
[102] http://www.everex.es/ip.htm
[103] http://www.everex.es/ip.htm

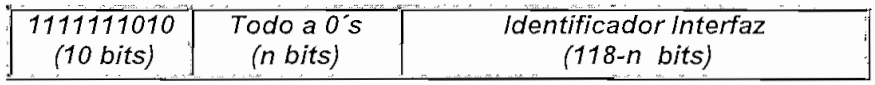


Fig. 2.19 Estructura de la dirección enlace local[104]

Las direcciones de enlace local son usadas para direccionar un solo enlace, para diferentes propósitos, como autoconfiguración de direcciones, descubrimiento de nodos vecinos, o cuando no existe un router.

La estructura de dirección ***Site-Local*** es la siguiente:

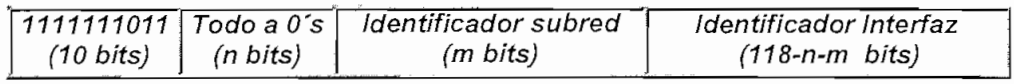


Fig. 2.20 Estructura de la dirección Site-local[105]

Las direcciones Site-Local se usan en grupos de redes que no disponen de una conexión a Internet, no necesitando un prefijo de dirección para su direccionamiento en Internet. En el momento en que el grupo se conecte a Internet, el prefijo de Site-Local será sustituido por un prefijo que identifique al grupo en la estructura global de Internet.

2.8.6 DIRECCIONES ANYCAST[106]

Una dirección IPv6 *anycast* es una dirección asignada a un grupo de interfaces, con la particularidad de que un paquete con una dirección anycast es llevada a sólo a un host, que será el más cercano según las técnicas de enrutamiento. Las direcciones *anycast* usan los mismos formatos definidos para direcciones unicast, con la diferencia de que el campo identificador de host estará todo en 0´s.

[104] http://www.everex.es/ip.htm
[105] http://www.everex.es/ip.htm
[106] http://a01-unix.gsyc.inf.uc3m.es/~baudoux/intro.htm

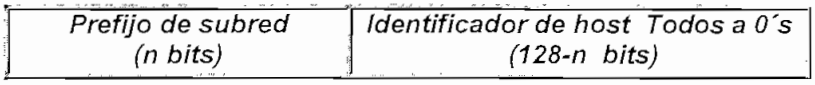


Fig. 2.21 Estructura de la dirección anycast[107]

Una dirección *anycast* no podrá nunca aparecer como dirección origen en un paquete IPv6, ni podrá ser asignada a ningún host. Las direcciones anycast sólo podrán ser asignadas a un *router*.

2.8.7 DIRECCIONES MULTICAST

Una dirección *multicast* es un identificador para un grupo de nodos. Un nodo puede pertenecer a cualquier grupo multicast. Las direcciones *multicast* tiene la siguiente estructura:

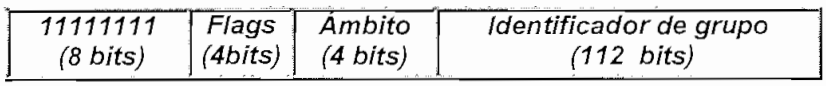


Fig. 2.22 Estructura de la dirección multicast[108]

- **Flags:** Este campo ocupa 4 bits 000T, donde 000 está reservado y
  - **T=0** indica direcciones asignadas permanentemente o conocidas, asignadas por una autoridad.
  - **T=1** indica que la dirección es de tránsito.

- **Ámbito:** Este campo ocupa 4 bits e indica el ámbito del grupo multicast; los valores que puede tomar son:
  - **0** Reservado
  - **1** Nodo Local

[107] http://a01-unix.gsyc.inf.uc3m.es/~baudoux/intro.htm

[108] http://a01-unix.gsyc.inf.uc3m.es/~baudoux/intro.htm

- 2 Enlace Local
- 3 Sin asignar
- 4 Sin asignar
- 5 Site Local
- 5 Sin asignar
- 6 Sin asignar
- 7 Sin asignar
- 8 Organización Local
- 9 Sin asignar
- A Sin asignar
- B Sin asignar
- C Sin asignar
- D Sin asignar
- E Global
- F Reservado

- **Identificador de grupo**

Este campo ocupa 12 bits, e identifica al grupo en el ámbito indicado, sea fijo o de transición. Las direcciones fijas tienen un significado independiente del ámbito que se indique.

2.8.7.1 Multicast con IP[109]

La mayoría de los protocolos de alto nivel de una red (como los protocolos de transporte de ISO, TCP o UDP) sólo proporcionan un servicio de transmisión *unicast*, es decir, los nodos de una red sólo son capaces de enviar paquetes de uno a otro en un momento dado.

Toda transmisión a través de un servicio unicast es inherentemente punto a punto. Si un nodo desea enviar la misma información a muchos destinos por

[109] http://a01-unix.gsyc.inf.uc3m.es/~baudoux/intro.htm

medio de un servicio de transporte unicast, deberá llevar a cabo un **unicast replicado** para después enviar N copias de los datos a cada uno de los destinos.

Una mejor forma de transmitir datos desde un origen a varios destinos es proporcionar un servicio de transporte **multicast**. De esta forma, un nodo puede enviar datos a varios destinos haciendo simplemente una llamada al servicio de transporte.

Para aquellas aplicaciones que implican que un nodo envíe a varios receptores, el multicast es claramente un paradigma de programación más natural que el unicast. Sin embargo, las ventajas del multicast son más que lógicas. Muchos medios de transmisión (como Ethernet) proveen un soporte para multicast y broadcast a nivel físico y de acceso al medio. Cuando se implementa un servicio multicast sobre una red, se produce un fuerte incremento de las prestaciones.

- **Aplicaciones del Multicast**

El Multicast es aconsejable porque permite la construcción de aplicaciones verdaderamente distribuidas, y porque permite una importante optimización del rendimiento sobre las transmisiones unicast. Hay un gran número de aplicaciones para multiconferencia audio y video en tiempo real, las cuales pueden hacer un buen uso de un servicio multicast.

- **Multicast sobre redes IP**

IP Multicast es un protocolo que sirve para transmitir datagramas IP desde un origen a varios destinos en redes de área local o extensa que funcionen bajo la pila de protocolos TCP/IP. Lo que el protocolo IP proporciona básicamente es un servicio de transmisión *unicast*. Es decir, el actual estándar de IP facilita la transmisión de datagramas desde un único origen hasta un único destino.

Sin embargo, las investigaciones hechas demuestran que no son necesarias modificaciones muy severas para añadir una base para el encaminamiento en

multicast a IP. El protocolo de encaminamiento IP Multicast facilita el envío de datagramas de un origen a un número arbitrario de destinos en una red grande y heterogénea como es Internet.

- **Multicast y redes ATM**

Las aplicaciones de multiconferencia audio y video consumen gran parte del ancho de banda y requieren una latencia extremadamente baja del servicio de multicast de la red. Para solucionar el problema del ancho de banda puede hacerse que IP corra bajo más soportes, pero permanece un serio problema de latencia en las redes IP. Para mejorar la eficiencia, puesto que IP admite paquetes muy grandes, un paquete de tiempo real pequeño puede ser incorporado dentro de un gran paquete. Se está investigando para solucionar éste problema en las redes ATM.

2.9 ENCAMINAMIENTO (ROUTING)[110]

En Internet, cada nodo tiene una tabla de encaminamiento que contiene información sobre otros nodos de la red, de forma que los nodos pueden comunicar unos con otros tomando como referencia la tabla. El encaminamiento en IPv6 trata las direcciones de una red como un conjunto de identificadores, cada red requiere una entrada en la tabla de encaminamiento.
Debido al crecimiento de Internet, el encaminamiento se hace menos manejable con respecto a la eficiencia y requerimientos de memoria, pues aumenta el número de direcciones IP.

Para mantener la actual inversión en protocolos y aplicaciones de Internet, el encaminamiento de IPv6 es casi idéntico al de IPv4. Todo esto necesitará una transición muy controlada para que IPv6 sea operativo con los algoritmos que se usan en IPv4 (RIP, OSPF, etc).

---

[110] http://a01-unix.gsyc.inf.uc3m.es/~baudoux/intro.htm

Las diferencias de encaminamiento son:

- *La longitud de las direcciones*, 128 bits en lugar de 32, lo que permite más niveles de jerarquía para reducir el tamaño de la tabla de encaminamiento y, como consecuencia, más eficiencia con menos memoria.

- *Extensiones de encaminamiento que soportan nuevas funcionalidades de encaminamiento.* Esto permite varias características nuevas:

    - *Selección de ruta*: Una opción que permite a la máquina origen listar los nodos intermedios necesarios para alcanzar el destino.

    - *Máquinas móviles*: También llamadas plug and play. Esta función permitirá conectar una máquina a la red y poder alcanzar y ser alcanzada sin necesidad de configuraciones manuales. Las direcciones IP serían automáticamente asignadas y las tablas adecuadas debidamente actualizadas.

    - *Redirección automática*: El destino puede responder a la dirección origen invirtiendo la secuencia de direcciones, eliminando así el proceso de encaminamiento.

2.10 AUTOCONFIGURACIÓN DE DIRECCIONES[111,112,113]

2.10.1 UNA NECESIDAD CRECIENTE

Los datos de las redes son cada vez más complejos, y la necesidad de eliminar algunas dificultades convierte al "Plug and Play" (Servicio inmediato) en algo cada vez más imprescindible, el usuario no tiene que conocer en detalle la arquitectura

[111] RFC 1752 The Recommendation for the IP Next Generation Protocol
[112] RFC 2460 Internet Protocol, Version 6 (IPv6) Specification. S. Deering, R. December 1998.
[113] RFC 2462 IPv6 Stateless Address Autoconfiguration. S. Thomson, T. Narten. December 1998.

de la red, ni saber configurar el software de red de su estación de trabajo. En el caso ideal, cualquier usuario debería ser capaz de desembalar su nuevo ordenador, conectarlo a la red local y verlo funcionar sin la necesidad de introducir cierta información de esta red. Ciertas preocupaciones de seguridad pueden limitar este nivel de transparencia de autoconfiguración de direcciones en algunos entornos, pero deben existir mecanismos para soportar cualquier automatización en el que el entorno local estaría de acuerdo.

2.10.2 EXIGENCIAS

La primera exigencia de la operación "Plug and Play" es que una estación pueda ser capaz de adquirir una dirección de manera dinámica, ya sea cuando está conectada por primera vez a una red, o cuando la estación necesite ser reconfigurada por traslado o por que la identidad de la red ha sido modificada. Existen también otras funciones que necesitan un entorno de "Plug and Play". La mayoría de ellas se deben hacer fuera del protocolo IPv6, pero el protocolo de autoconfiguración de direcciones de una estación será ejecutado por IPv6.

2.10.3 TIPOS DE DIRECCIONES AUTOCONFIGURABLES

Una estación IPv6 puede autoconfigurar dos tipos de direcciones:

- Las direcciones de intra-enlace (*intra-link scope address*).
- Las direcciones de Inter-enlace (*Inter-link scope address*)

Una dirección de intra-enlace es auto configurable en ausencia de encaminador, mientras que una dirección de Inter-enlace es auto configurable cuando un encaminador está presente en el enlace.

2.10.4 PROCEDIMIENTOS DE FORMACIÓN DE LAS DIRECCIONES[114]

- **Direcciones Intra-enlace:** Solo existe una manera para formar una dirección intra-enlace. Al inicializar la interfaz, una estación crea su dirección de intra-enlace concatenando un prefijo de intra-enlace a una ficha (token) que es única para el enlace. Típicamente, la definición de una ficha es independiente de la capa de enlace. Por ejemplo, en el caso de una estación conectada a una red IEEE 802, la ficha es la dirección IEEE 802 del interfaz.

- **Direcciones Inter-enlace:** Existen dos maneras para crear una dirección Inter-enlace. En el primer mecanismo una estación obtiene su dirección de Inter-enlace concatenando un prefijo de red indicado por un "Router Advertisement" a una ficha única por enlace. El otro mecanismo disponible para las estaciones es utilizar el protocolo de configuración dinámica de las estaciones para IPv6 (Dynamic Host Configuration Protocol – DHCP). La elección del protocolo a utilizar es propuesta por el encaminador, y la elección es configurable por el administrador del sistema.

  El primer proceso de formación de la dirección de inter-enlace conviene para entornos donde ninguna gestión administrativa es deseable.

  Este protocolo está concebido especialmente para permitir una configuración sencilla de las direcciones. DHCPv6 es un protocolo más complejo que permite una asignación flexible de direcciones bajo el control del administrador del sistema. Este protocolo necesita sobre todo un gestor de sistemas (servidor y base de datos) importante.

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

2.10.5 PROCEDIMIENTOS PARA FORMAR DIRECCIONES[115]

Una estación mantiene una lista de direcciones por interfaz. Al menos, la lista contendrá una dirección de intra-enlace que puede formar automáticamente la estación cuando un interfaz se inicializa. Si un encaminador está conectado al enlace, la lista incluirá también las direcciones de inter-enlace, formadas por prefijos de subred reclamados ya sea por peticiones a los encaminadores de advertisement o haciendo llamadas a DHCPv6. Las direcciones de inter-enlace también pueden configurarse manualmente.

2.10.6 CONFIGURACIÓN DE LAS ESTACIONES[116]

Una estación puede mantener una lista de variables de configuración por interfaz:

- **Dirección:** Una dirección unicast IPv6 válida en este interfaz. Por defecto nada.

- **Tiempo de vida (Life Time):** El tiempo de vida en el cual la dirección es válida, medido en segundos. Por defecto tiempo infinito.

  Una dirección de intra-enlace y todas las direcciones configuradas manualmente tienen su tiempo de vida puesto a infinito. Una estación debe permitir configurar la variable siguiente de la lista (por el administrador del sistema y para cada interfaz).

- **Perform_Auto_Address:** Si su valor es verdadero (TRUE) la estación deberá proceder a una configuración de direcciones automática, y no realizar ninguna autoconfiguración. Por defecto TRUE.

---

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

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

2.10.7 CONFIGURACIÓN DE ENCAMINADORES[117]

Un encaminador debe ser configurado por el administrador del sistema; así se puede controlar la elección del mecanismo utilizado para la configuración de estaciones (con respecto a sus direcciones autoconfigurables).

- **Perform_Auto_Address:** Si y solo si esta variable está puesta a TRUE el encaminador manda una extensión de prefijo de dirección a todos los encaminadores de advertisement.

2.10.8 PROCEDIMIENTOS DE AUTOCONFIGURACIÓN DE DIRECCIONES DE ESTACIONES[118]

Una estación debe seguir los siguientes procedimientos para cada interfaz cuando se arranca o cuando debe inicializarse un interfaz:

- Cuando una estación arranca o en cualquier momento en que no tiene ninguna dirección, esta estación produce una dirección de intra-enlace y la añade a su lista de direcciones.

- La estación debe mandar una petición al encaminador (Router Solicitation) para realizar (o verificar) lo más rápidamente posible sus direcciones de inter-enlace. Cuando es solicitado un encaminador de advertisement, la estación debe tratar la configuración de direcciones de la siguiente manera:

    - Si existe una extensión de prefijo de dirección, la estación forma o comprueba sus direcciones de Inter-enlace autónomas. En caso contrario, esto implica que se debe utilizar el protocolo DHCPv6 para la autoconfiguración de direcciones. Si no existe ninguna dirección por el interfaz, la estación pone en marcha una petición al servidor

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

[118] RFC 2462 IPv6 Stateless Address Autoconfiguration. S. Thomson, T. Narten. December 1998

de DHCPv6 para adquirir una nueva dirección. Si por cualquier motivo, DHCPv6 no lo consigue, la estación vuelve a utilizar una dirección de intra-enlace o una dirección de inter-enlace configurada manualmente hasta que logre la petición al servidor.

2.10.9 FORMACIÓN DE UNA DIRECCIÓN IPv6 A PARTIR DE UNA DIRECCIÓN IEEE 802[119]

Una estación puede formar una dirección IPv6 para una interfaz concatenando un prefijo de subred de 80 bits con una dirección IEE802 de 48 bits del interfaz como sigue:

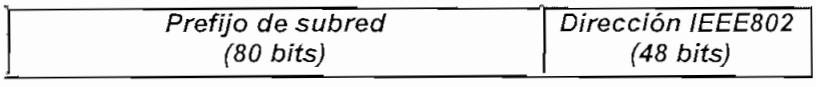


Fig. 2.23 Dirección IPv6 a partir de una dirección IEEE802[120]

En caso de un prefijo de intra-enlace, el prefijo de subred está bien definido, mientras que, en caso de un prefijo de inter-enlace, el prefijo de subred es configurable.

2.11 CONTROL DE FLUJO[121]

Para soportar aplicaciones que requieran algún grado de capacidad de tratamiento, retardo o prioridad, el campo Etiqueta de Flujo de la cabecera IPv6 puede ser usado por una máquina para, marcar aquellos paquetes que requieran un especial manipulamiento por routers IPv6, como los siguientes:

- Calidad de servicio no por defecto
- Servicio en tiempo real

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

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

[121] RFC 1809 Using the flow Label Field in IPv6

Un *flujo* es una secuencia de paquetes enviados desde un origen particular a un destino (unicast o multicast) para el cual el emisor desea un especial manipulamiento en la intervención de los routers.

La naturaleza de estas manipulaciones pueden ser comunicadas a los routers:

- Por un protocolo de control, como por ejemplo un protocolo de reserva de recursos.
- Por información contenida dentro de los mismos paquetes pertenecientes al flujo.

Un flujo es identificado por una dirección fuente y el identificador de flujo puesto a un valor distinto de cero, ya que los paquetes que no pertenecen a un flujo llevan el identificador de flujo puesto a cero.

2.12 SEGURIDAD IPv6[122]

Primero definiremos algunos términos necesarios:

- **Autenticidad**: Propiedad de conocer que la información recibida sea la misma que la información enviada y que el emisor es realmente quien asegura.

- **Integridad**: Propiedad de asegurar que la información es transmitida desde el emisor al destino sin detectarse alteraciones.

- **Confidencialidad**: Propiedad de mantener comunicaciones confidenciales de modo que los participantes involucrados puedan establecer comunicación sin que otros elementos ajenos a ellos sepan quienes son.

[122] Internet-draft-ietf-dhcpwg-dhcpv6-00

- **Cifrado:** Mecanismo utilizado comúnmente para proveer confidencialidad.

- **No repudio:** Propiedad de que un receptor sea capaz de probar que el emisor envió realmente información, aún cuando el emisor pudiera negar posteriormente haber enviado dicha información.
- **SAID:** Acrónimo de "Identificador de Asociación de Seguridad".

- **Asociación de Seguridad:** Conjunto de información de seguridad referente a una conexión de red dada. Esta incluye generalmente la clave criptográfica, tiempo de vida de la clave, algoritmo, forma del algoritmo, nivel de sensibilidad, (por ejemplo, no clasificada, secreto, propietario), que clase de servicio de seguridad se proporciona y posiblemente alguna otra información.

- **Análisis de tráfico:** Una clase de ataque de red es aquel en el cual el atacante es capaz de hacer deducciones acerca de la misma analizando sólo los patrones de tráfico de red (como frecuencia de transmisión, con quien se habla, tamaño de paquetes, identificador de flujo utilizado, etc.).

El actual IPv4 presenta algunos problemas de seguridad, básicamente carece de confidencialidad y de métodos de autenticación por debajo del nivel de aplicación. IPv6 ofrece la posibilidad de solucionar esto proporcionando dos opciones integradas que traen consigo seguridad. Estas dos opciones pueden ser utilizadas solas o juntas, dependiendo de las necesidades del usuario.

2.12.1 OPCIONES DE SEGURIDAD[123]

- **Cabecera de autenticación IPv6:** Proporciona autenticación e integridad, pero no confidencialidad. La opción será independiente de algoritmos y soportará varias técnicas de autenticación. El uso de "keyed MD5" (algoritmo estándar de autenticación, es una función segura que convierte

[123] Internet-draft-ietf-dhcpwg-dhcpv6-00

una secuencia de datos larga en un resumen de longitud fija) se ha propuesto para asegurar interoperabilidad dentro de la Internet, y también suprimir un cierto número de ataques.

Esta protección (a nivel de Internet) daría a los niveles superiores una autenticación de las máquinas origen, proporcionándoles la protección mínima necesaria que ahora no tienen.

La ventaja de proporcionar todo lo anterior salvo confidencialidad es que este mecanismo sería exportable por vendedores en aquellos países como Estados Unidos, donde se restringe la exportación de algoritmos de confidencialidad.

- **Cabecera encapsulada de Seguridad:** Esta opción dará integridad y confidencialidad ausentes en la opción de cabecera de autenticación de IPv6. Es al mismo tiempo flexible e independiente de algoritmos, el algoritmo "DES" (estándar de encripción de datos) ha sido propuesto como el estándar con el fin de conseguir interoperabilidad en toda la Internet. Sin embargo, este mecanismo probablemente no será tan exportable como la cabecera de autenticación, pero el uso de DES como estándar debería ayudar.

2.12.2 OBJETIVOS DE DISEÑO[124]

El objetivo principal es garantizar que IPv6 tenga mecanismos de seguridad sólidos disponibles para usuarios que deseen seguridad. Estos mecanismos están diseñados de tal manera que los usuarios de Internet que no los empleen no se vean afectados.

---

[124] Internet-draft-ietf-dhcpwg-dhcpv6-00

Se pretende que estos mecanismos sean algoritmos independientes de forma que los algoritmos de cifrado puedan ser alterados sin afectar otras partes de la implementación.

Los algoritmos estándar por defecto (por ejemplo: MD5 con clave, DES) han sido seleccionados para garantizar interoperabilidad en la Internet global. Los algoritmos seleccionados son los mismos que los algoritmos estándar por defecto utilizados en SNMPv2. Los mecanismos de seguridad de IPv6 deberían ser así útiles al imponer una variedad de políticas de seguridad.

2.12.3 MECANISMOS DE SEGURIDAD DE IPv6[125]

Hay dos mecanismos de seguridad en IPv6:

- La Cabecera de Autenticación, proporciona integridad y autenticidad sin confidencialidad.

- El Encapsulating Security Payload (encapsulamiento de seguridad de datos), que dependiendo del algoritmo y modo, podría proporcionar integridad, autenticidad, y siempre confidencialidad.

Los mecanismos de IPv6 no proveen seguridad contra un número de ataques de análisis de tráfico. Sin embargo, hay varias técnicas que podrían ser utilizadas para proporcionar protección contra el análisis de tráfico.

2.12.3.1 Cabecera de autenticación

La cabecera de autenticación de IPv6 busca proporcionar integridad y autenticidad para los datagramas IPv6. Esto se hace computando una función de autenticidad criptográfica sobre el datagrama IPv6 y empleando una clave de autenticidad secreta en el cálculo. El emisor computa la información de

[125] Internet-draft-ietf-dhcpwg-dhcpv6-00

autenticidad exactamente antes de enviar el paquete IPv6 autenticado y el receptor verifica la información autenticada con la recibida. Hay ciertos campos que son omitidos en el cálculo de autenticidad debido a que cambian durante el tránsito como el campo Hop Limit o Límite de Saltos, decrementado en cada paso. Sin embargo, la omisión del campo Hop Limit no afecta a la seguridad. Algunos algoritmos de autenticación podrían proporcionar no repudio (por ejemplo, algoritmos asimétricos en los que tanto las claves del emisor como del receptor se utilizan en el cálculo de autenticidad) utilizados con la cabecera de autenticidad, pero no es necesariamente suministrado por todos los algoritmos de autenticidad que pueden utilizarse con la Cabecera de Autenticación.

El algoritmo de autenticación por defecto es el MD5 con clave, que se ajusta a todos los algoritmos simétricos que no proporcionan no repudio. La protección del análisis de tráfico y la confidencialidad no son suministradas por la Cabecera de Autenticación.

La Cabecera de Autenticación de IPv6 mantiene información de autenticación para su datagrama IPv6. Esta información de autenticación se calcula utilizando todos los campos del datagrama IPv6 que no van a cambiar en el tránsito.

El uso de la Cabecera de Autenticación aumentará los costes de procesamiento de protocolo de IPv6 en los sistemas que lo utilicen, así como la latencia de las comunicaciones. El aumento de latencia es principalmente debido al cálculo de la información de autenticación por parte del emisor y el cálculo y comparación de la información de autenticación por el receptor de cada datagrama IPv6 contenido en una Cabecera de Autenticación.

La cabecera proporciona una seguridad más fuerte que la existente en la mayoría de la actual Internet y no debería afectar a la exportabilidad ni aumentar significativamente el coste de implementación. Aunque la cabecera puede ser instrumentada como una medida de seguridad empleando los nombres exactos de las máquinas de una red, este modo de operación no es seguro. En lugar de

eso, la Cabecera de Autenticación debería ser utilizada también desde el origen al destino final.

Todas las máquinas que soporten IPv6 tienen que implementar la Cabecera de Autenticación de IPv6 con al menos el algoritmo MD5 con unas claves de 128 bits. Se puede implementar otros algoritmos de autenticación además del MD5 con clave.

2.12.3.2 Encapsulating Security Payload[126]

El Encapsulating Security Payload (ESP) de IPv6 trata de dar integridad, autenticación y confidencialidad a los datagramas IPv6. Esto se hace por encapsulamiento, ya sea de un datagrama IPv6 completo o solamente información de protocolo de la capa superior dentro del ESP, cifrando la mayor parte del contenido del ESP, para concatenar después de una nueva cabecera IPv6 sin cifrar al ya cifrado ESP. Esta cabecera IPv6 no cifrada se utiliza para llevar los datos protegidos a través de la red. El receptor del datagrama no cifrado retira y descarta la cabecera IPv6 y sus opciones no cifradas, descifra el ESP, procesa y después elimina las cabeceras de ESP, trata el (ahora descifrado) datagrama original IPv6 o los datos de un protocolo de nivel superior, como se indica en las especificaciones del protocolo IPv6.

Hay dos modos dentro de ESP:

- El primer modo, conocido como IP-mode, encapsula y completa el datagrama IP dentro de la cabecera de ESP.

- El segundo modo conocido como Transport-mode, generalmente encapsula un UDP o TCP enmarcándolos dentro de IP.

[126] Internet-draft-ietf-dhcpwg-dhcpv6-00

ESP trabaja entre máquinas, entre una máquina y una entrada de seguridad, o entre entradas de seguridad. El soporte para entradas de seguridad permite que haya redes fiables detrás de una entrada de seguridad para omitir el cifrado y de ese modo evitar el trabajo y costes monetarios del cifrado, mientras que ofrece confidencialidad para tráfico transitando por segmentos de red no fiables. Cuando ambas máquinas implementan directamente ESP y no intervienen entradas de seguridad, entonces se puede emplear el Transport-mode (donde sólo es cifrada la información de protocolo de la capa superior (TCP o UDP) no la cabecera de IPv6). Este modo reduce tanto el ancho de banda consumido como los costes de procesamiento de protocolo para usuarios que no necesiten mantener la confidencialidad del datagrama IPv6 al completo. ESP trabaja tanto con tráfico unicast como multicast.

- **Impactos de la función de ESP**: El encapsulamiento de seguridad utilizado por ESP puede impactar notablemente la función de la red en sistemas participantes, pero no debería influir negativamente en encaminadores u otros sistemas intermedios que no participen en la asociación de ESP particular.

  El procesamiento de protocolo en sistemas participantes será más complejo cuando se utilice el encapsulamiento de seguridad, ambos requieren más tiempo y más potencia de procesamiento.

  El uso del cifrado también aumentará la latencia de las comunicaciones. La latencia aumentada es principalmente debida al cifrado y descifrado requerido por cada datagrama IPv6 contenido en un ESP. El coste preciso de ESP variará con las especificaciones de la implementación, incluyendo el algoritmo de cifrado, tamaño de clave y otros factores.

  Las implementaciones hardware de los algoritmos de cifrado son recomendables cuando se desee una gran productividad. Debido al impacto de las funciones, los usuarios que no requieran confidencialidad

preferirán probablemente utilizar la cabecera de Autenticación de IPv6 en lugar de ESP.

Para interoperar a través de la Internet a nivel mundial, todas las implementaciones de Encapsulating Security Payload de IPv6 soportan el uso del Data Encryption Standard (DES).

Otros algoritmos de confidencialidad y modos pueden ser implementados además de este algoritmo y modo. La exportación de métodos de cifrado y uso de los mismos está regulado en algunos países.

2.12.3.3 Combinando mecanismos de seguridad[127]

En algunos casos la Cabecera de autenticación de IPv6 puede combinarse con el IPv6 Encapsulating Security Protocol para obtener las propiedades de seguridad deseadas. La Cabecera de Autenticación proporciona siempre integridad y autenticación y puede incluir no repudio si se usa con ciertos algoritmos de autenticación.

El Encapsulating Security Payload proporciona siempre integridad y confidencialidad y puede proveer autenticación si se utiliza ciertos algoritmos de autenticación y cifrado.

Añadiendo la Cabecera de Autenticación a un primer datagrama IPv6 para encapsular aquel datagrama mediante el Encapsulating Security Protocol podría ser aconsejable para usuarios que deseen tener una integridad más fuerte, autenticación, confidencialidad, y quizás también no repudio. Cuando se combinan los dos mecanismos, la colocación de la Cabecera de Autenticación de IPv6 aclara que parte de la información está siendo autenticada.

---

[127] Internet-draft-ietf-dhcpwg-dhcpv6-00

2.12.3.4 Otros mecanismos de seguridad

La protección del análisis de tráfico no es facilitada por ninguno de los mecanismos de seguridad descritos anteriormente. No está claro que la protección significativa contra el análisis de tráfico pueda ser proporcionada económicamente en la Capa Internet y parece que pocos usuarios de Internet son conscientes acerca del análisis de tráfico.

Una técnica es enviar tráfico falso para aumentar el ruido en la información para el análisis de tráfico.

2.12.4 ADMINISTRACION DE CLAVES[128]

IPv6 pretende sostener la llamada administración de claves "in-band" donde la información de administración de claves se lleva en una cabecera de IPv6 distinta. En lugar de eso se utilizará principalmente la llamada administración de claves "out-of-band", donde la información de administración de claves la llevará un protocolo de capas superiores (como UDP o TCP) en algún número de puerto especificado.

Esto permite aclarar el desacople del mecanismo de administración de claves de otros mecanismos de seguridad, y de ese modo se permite una nueva y mejorada sustitución de métodos de administración sin tener que modificar las implementaciones de los otros mecanismos de seguridad.

2.12.4.1 Distribución manual de claves[129]

La forma más simple de administración de claves es la manual, donde una persona configura manualmente cada sistema con su propia clave y con las claves de otros sistemas de comunicación.

---

[128] Internet-draft-ietf-dhcpwg-dhcpv6-00

[129] Internet-draft-ietf-dhcpwg-dhcpv6-00

Esta es una práctica habitual en entornos estáticos, pequeños pero no de gran escala, es útil en muchos entornos a corto plazo, pero a medio o largo plazo no es un enfoque viable.

Por ejemplo, dentro de una LAN pequeña es totalmente práctico configurar manualmente claves para cada sistema. Dentro de un único dominio administrativo es útil configurar las claves para cada encaminador de modo que la información de encaminamiento quede protegida y para reducir el riesgo de que un intruso asalte un encaminador.
Otro caso puede ser una organización que tenga un firewall de cifrado entre la red interna y la Internet en cada una de sus plantas y se conecten dos o más plantas a través de la Internet.

En este caso, el firewall de cifrado podría selectivamente cifrar tráfico por otras plantas dentro de la organización utilizando una clave configurada manualmente, siempre y cuando no cifre tráfico con otros destinos. También podría ser apropiado cuando solamente se necesita garantizar ciertas comunicaciones seleccionadas.

2.12.4.2 Algunas Técnicas de Administración de Claves Existentes[130]

Hay varios algoritmos de administración de claves descritos en literatura de tipo público.

Needham y Schroeder han propuesto un algoritmo de administración de claves que confía en un sistema de distribución de claves centralizado.

Más recientemente, Diffie y Hellman han ideado un algoritmo que no requiere de un sistema de distribución de claves centralizado. Lamentablemente, la técnica original de Diffie-Hellman es vulnerable, sin embargo, esta vulnerabilidad puede ser mitigada usando la firma de claves (firma digital) para la autentificación.

[130] Internet-draft-ietf-dhcpwg-dhcpv6-00

2.12.4.3 Distribución automatizada de claves[131]

La distribución y uso extendido de seguridad de IPv6 requerirá un protocolo de distribución de claves acoplable a la Internet estándar. Idealmente, tal protocolo sostendría un número de protocolos en la pila de protocolos de Internet, no sólo en la seguridad de IPv6.

Hay trabajo en camino dentro del IETF para añadir claves de máquinas asignadas al Sistema de Nombres de Dominio (DNS). Las claves de DNS permiten que la parte original pueda autenticar mensajes de administración de claves con la otra parte de administración de claves utilizando un algoritmo asimétrico. Entonces, las dos partes tendrían un canal de comunicaciones autenticable que podría emplearse para crear una clave de sesión compartida.

Hay dos enfoques de claves para IPv6.

- El primer enfoque, llamado clave host-to-host, tiene a todos los usuarios que comparten la misma clave en la máquina 1 para el uso en tráfico destinado a todos los usuarios en la máquina 2.

- El segundo enfoque, llamado clave user-to-user, permite al usuario A en la máquina 1 tener una clave de sesión única con el usuario B en la máquina 2 que no está compartida con otros usuarios en host 1.

En muchos casos, un sistema de computación único tendrá al menos dos usuarios A y B mutuamente desconfiados (que no confían el uno en el otro). Cuando se utiliza la clave host-to-host y existen usuarios mutuamente desconfiados, es posible por parte del usuario A averiguar la clave host-to-host por medios conocidos. Una vez que el usuario A ha obtenido impropiamente la clave en uso, el usuario A puede entonces o bien leer el tráfico cifrado del usuario

[131] Internet-draft-ietf-dhcpwg-dhcpv6-00

B o bien falsificar tráfico del usuario B. Cuando se utiliza un clave host-to-host, este tipo de ataques de un usuario al tráfico de otro usuario no es posible. Por tanto, se deduce que las claves host-to-host deberían estar presentes en todas las instrumentaciones de IPv6.

2.12.4.4 Distribución de Claves de Multicast

La Distribución de Claves de Multicast es un área de investigación activa en la literatura. Para grupos de multicast que tienen relativamente pocos miembros, la distribución manual de claves o el uso de múltiples algoritmos de distribución de claves unicast existentes, como los Diffie-Hellman modificados, parecen posibles. Para muchos grupos, las nuevas técnicas de adaptación son necesarias.

2.12.4.5 Requisitos de Administración de claves de IPv6[132]

Se definen requisitos de administración de claves para todas las instrumentaciones de IPv6. Esto se aplica igualmente a la Cabecera de Autenticación de IPv6 y al Encapsulating Security Payload de IPv6.

- Todas las implementaciones de IPv6 tienen que soportar una administración de claves manual.

- Todas las implementaciones de IPv6 deberían soportar un protocolo de administración de claves estándar de Internet una vez que este último sea definido.

- Todas las implementaciones de IPv6 tienen que permitir la configuración de uso de claves user-to-user para el tráfico originado en el mismo sistema y poder permitir adicionalmente la configuración de claves host-to-host para el tráfico originado en ese sistema como una característica añadida para

[132] Internet-draft-ietf-dhcpwg-dhcpv6-00

hacer la distribución manual de claves más fácil y dar al administrador del sistema más flexibilidad.

- Un dispositivo que cifre o autentique paquetes de IPv6 originados en otros sistemas, como por ejemplo un cifrador IP dedicado o un gateway cifrador, por lo general no puede proporcionar claves user-to-user para el tráfico originado en otros sistemas. Por tanto, tales sistemas tienen que implementar claves host-to-host para tráfico originado en otros sistemas.

- El método por medio del cual las claves son configuradas en un sistema particular quedará definido en la implementación. Un archivo plano que incluya identificadores de asociación de seguridad y los parámetros de seguridad, incluyendo las claves, es un ejemplo de un posible método para la distribución manual de claves.

- Un sistema IPv6 tiene que dar pasos razonables para proteger las claves y otra información de seguridad, ya que todas la seguridad radica en las claves.

2.12.5 USO[133]

Existen varios mecanismos de seguridad suministrados por IPv6 en diferentes entornos y aplicaciones para dar al implementador y al usuario una mejor idea de cómo estos mecanismos puede utilizarse para reducir riesgos de seguridad.

2.12.5.1 Uso con firewalls

Los firewalls son habituales en la actual Internet. Los dos mecanismos de IPv6 pueden utilizarse para aumentar la seguridad suministrada por los firewalls.

[133] Internet-draft-ietf-dhcpwg-dhcpv6-00

Los firewalls usados con IPv6 deben poder analizar la cabecera para determinar el protocolo de transporte (p.e. UDP o TCP) en uso y el número de puerto para esos protocolos. La función del Firewall no debe verse afectada significativamente por el uso IPv6, debido a que las reglas de formato de cabecera de IPv6 hacen un análisis fácil y rápido.

Los firewalls pueden utilizar la Cabecera de Autenticación para asegurarse de que la información (p.e. emisor, destino, protocolo de transporte, número de puerto) que se utiliza en decisiones de control de acceso es correcta y auténtica. La autenticación podría estar desempeñada no solamente dentro de una organización o campus sino también con sistemas remotos a través de Internet mediante conexiones "end to end". Este uso de la Cabecera de Autenticación con IPv6 proporciona mucha más seguridad que la que facilita IPv4.

Organizaciones con dos o más plantas que se interconectan utilizando un servicio de IP comercial podrían desear utilizar un firewall para el cifrado selectivo. Si entre cada planta de una determinada empresa y su proveedor de servicios IP se situase un firewall de cifrado éste podría proporcionar un túnel de IP de cifrado entre todas las plantas de la empresa; podría también cifrar tráfico entre dicha empresa y sus suministradores, clientes y otros afiliados. El tráfico con el NIC, con el archivo público de Internet, o alguna otra organización no podría ser cifrado debido a la no disponibilidad de un protocolo de administración de claves estándar o al deseo de facilitar unas comunicaciones mejoradas, asi como también mejores funciones de red, y aumento de la conectividad. Tal práctica puede proteger fácilmente el tráfico sensible de la organización de posibles caídas y modificaciones.

Algunas organizaciones (como gobiernos) podrían desear utilizar completamente un firewall de cifrado para dar lugar a una red virtual protegida sobre el servicio comercial de IP. La diferencia entre esto y un dispositivo de cifrado de IPv6 es que un firewall dedicado al cifrado proveería filtraciones del tráfico descifrado así como ofrecería cifrado de paquetes de IP.

2.12.5.2 Uso con IPv6 Multicast[134]

En los últimos años, la Multicast Backbone (MBONE) ha crecido rápidamente. Las reuniones de IETF y otras conferencias son ahora regularmente de tipo multicast con audio en tiempo-real, video, y whiteboards. Mucha gente está utilizando ahora aplicaciones de teleconferencia basadas en IP MULTICAST en la Internet o en redes internas privadas. Por tanto, es importante que los mecanismos de seguridad en IPv6 sean apropiados para su uso en un entorno donde el tipo multicast es el caso general.

Los Security Association Identifiers (SAIDS), utilizados en mecanismos de seguridad de IPv6 están orientados al receptor, haciéndolos apropiados para uso en multicast IP, lamentablemente, los protocolos de distribución de claves multicast actualmente publicados no se adaptan bien. Sin embargo, hay una investigación activa en esta área. Como paso interno, un grupo multicast podría utilizar repetidamente un protocolo seguro de distribución de claves unicast para distribuir la clave a todos los miembros o el grupo podría prearrancar claves utilizando una distribución de claves manual.

2.12.6 CONSIDERACIONES DE SEGURIDAD[135]

Los usuarios deben entender que la calidad de la seguridad suministrada por los mecanismos de IPv6 depende completamente de la fortaleza de los algoritmos criptográficos implementados, la robustez de la clave que se utiliza, la implementación correcta de los algoritmos criptográficos, la seguridad del protocolo de administración de claves, la implementación correcta de IPv6 y los distintos mecanismos de seguridad de todos los sistemas que intervienen .

La seguridad de la implementación está en parte relacionada con la seguridad del sistema operativo que encarna las implementaciones de seguridad. Por ejemplo,

---

[134] Internet-draft-ietf-dhcpwg-dhcpv6-00

[135] Internet-draft-ietf-dhcpwg-dhcpv6-00

si el sistema operativo no mantiene la confidencialidad de las claves cifradas privadas, entonces el tráfico que utilice dichas claves no va a ser seguro. Si cualquiera de éstas fuera incorrecta o insuficientemente segura, poca o ninguna seguridad real tendría el usuario. Debido a que diferentes usuarios del mismo sistema podrían no confiar unos en otros, cada usuario o cada sesión debería trabajar generalmente con claves separadas. Esto causará que aumente el trabajo requerido para criptoanalizar el tráfico, ya que no todo tráfico utilizará la misma clave.

Ciertas propiedades de seguridad (como protección del análisis de tráfico) no las pueden aportar estos mecanismos. Los usuarios tienen que considerar cuidadosamente que propiedades de seguridad quieren tomar para garantizar que sus necesidades se encuentran en estos u otros mecanismos.

Ciertas aplicaciones (como correo electrónico) necesitan probablemente mecanismos de seguridad específicos para cada aplicación. Estos mecanismos están fuera del alcance de la Arquitectura de Seguridad de IPv6.

2.12.7 DNS PARA IPv6[136]

La resolución de direcciones DNS funciona de forma similar a los mecanismos vistos para IPv4. Además, se han añadido algunas características que son:

- Un nuevo tipo de petición para soportar direcciones IPv6
- Un nuevo dominio
- Todos los procesos adicionales que se requerían para localizar direcciones IPv4 son redefinidos para localización de direcciones tanto IPv4 como IPv6

[136] http://www.everex.es/ip.htm

- La nueva petición se llama AAAA y es el tipo 28 en decimal. El formato de datos de AAAA es una dirección IPv6 de 128 bits

- Nuevo dominio que es el IP6.INT en el cual una dirección IPv6 se representa con un nombre en este dominio, la secuencia anterior a IP6.INT se codifica de forma inversa, por ejemplo la dirección IPv6 4321:0:1:2:3:4:567:89ab sería representada por b.a.8.9.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.0.1.2.3.4.

CAPITULO 3

MECANISMOS DE DE TRANSICIÓN DE IPv4 A IPv6

CAPÍTULO III

MECANISMOS DE TRANSICIÓN DE IPv4 A IPv6

3.1 INTRODUCCIÓN[139]

La migración de IPv4 a IPv6 en un instante dado es imposible, debido al tamaño enorme de Internet y al gran número de usuarios de IPv4. Por otro lado muchas organizaciones empiezan a ser muy dependientes de la Internet para su trabajo diario, por lo que no se puede tolerar períodos largos de inactividad para el reemplazo del IP actual. Por lo tanto no podemos indicar cuando IPv4 deje de funcionar e IPv6 ingrese definitivamente.

Los dos protocolos pueden coexistir sin ningún problema, la migración de IPv4 a IPv6 será puesta en marcha nodo por nodo, usando procedimientos de autoconfiguración para eliminar la necesidad de configurar manualmente los host (ordenadores). Esto permitirá aprovechar inmediatamente de las múltiples ventajas de IPv6, mientras se mantiene la posibilidad de comunicarse con usuarios de IPv4.

Algunas características de IPv6 están diseñadas con el fin de simplificar la migración. Por ejemplo, las direcciones IPv6 se pueden derivar automáticamente de las direcciones IPv4, túneles IPv6 se pueden construir en las redes IPv4 y, por lo menos en la fase inicial, todos los nodos IPv6 utilizarán el "*dual stack*" o doble capa IP es decir utilizarán IPv4 e IPv6 al mismo tiempo.

Las metas dominantes de la migración son:

- Host IPv6 e IPv4 deben interoperar

[139] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997.

- Routers y hosts de IPv6 pueden ser distribuidos en Internet de un modo altamente difuso e incremental, con pocas interdependencias .
- La transición debería ser tan fácil y simple como sea posible para que los usuarios terminales, administradores del sistema y operadores de red puedan entenderla y ponerla en ejecución.

Un conjunto de mecanismos, llamado SIT (Transición de Internet Simple) ha sido implementado: incluye protocolos y reglas administrativas para simplificar la migración.

Las principales características del SIT son las siguientes:

- *Posibilidad de una transición progresiva y no traumática.* Los host y routers IPv4 instalados pueden ser mejorados a IPv6, uno a la vez y sin ser dependientes de otros hosts o routers ya mejorados.

- *Requisitos mínimos para ponerse al día.* El único requisito para mejorar hosts a IPv6 es que el servidor de DNS tenga que estar primero mejorado para manipular los registros de direcciones IPv6. No hay requisitos para la mejora de routers.

- *Simplicidad de Direccionamiento.* Cuando existiendo hosts o routers de IPv4 instalados se los mejora a IPv6, éstos pueden continuar utilizando su dirección IPv4. No necesitan ser asignados a nuevas direcciones.

- *Costos iniciales bajos.* Poco o ningún trabajo de preparación es necesario para mejorar la existencia de sistemas de IPv4 a IPv6, o para desplegar nuevos sistemas de IPv6.

Los mecanismos usados por el SIT incluyen:

- Una estructura de direccionamiento IPv6 que permita derivar direcciones IPv6 de las direcciones IPv4.

- Uso de la capa IP Dual en los hosts y routers durante la transición, es decir la presencia de pilas de IPv4 e IPv6 al mismo tiempo.

- Un mecanismo para "tunneling" de paquetes de IPv6 sobre infraestructuras de encaminamiento de IPv4. Esta técnica utiliza estructuras de direcciones IPv4, que elimina la necesidad de configuración de túnel en la mayoría de los casos.

- Un mecanismo opcional para la traducción de cabeceras de paquetes de IPv4 en IPv6, y las cabeceras de paquetes de IPv6 en IPv4. Esta técnica permite a nodos que implementan sólamente IPv6 interoperar con nodos que implementan sólamente IPv4.

El SIT garantiza que los hosts con IPv6 pueden interoperar con hosts IPv4 inicialmente en toda la red. Cuando la migración haya terminado esta interoperabilidad será localmente garantizada por un largo tiempo. Esto permitirá proteger las inversiones hechas en IPv4, los dispositivos simples que no se pueden poner al día a IPv6, por ejemplo impresoras de la red, servidores terminales seguirán funcionando con IPv4 hasta el final de su existencia.

La posibilidad de una migración gradual permite que cualquier fabricante integre IPv6 en routers, sistemas operativos y software de red cuando piensen que la implementación sea estable y en el momento que consideren más apropiado.

3.2 REDES DE DOBLE CAPA IP[140,141]

La característica importante de la transición es que permite a IPv6 ser añadido a la red después de un período de tiempo sin romper con la infraestructura de IPv4 existente. Esto es posible gracias a un esquema de transición *doble capa IP* el cual permite que a IPv6 se le añadan host, servidores DNS y routers, sin ningún

[140] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997.

cambio o ruptura en el soporte actual de IPv4. Una red de doble capa IP (IPv4 e IPv6) opera en un dominio simple de enrutamiento de Internet , el cual a su vez está conectado a los backbones de Internet regionales. Los nodos incluyen dos hosts de doble capa IP, así como dos hosts sólo con IPv4, más servidor DNS y dos routers de borde.

Un aspecto principal del plan de transición a IPv6 es la asignación de direcciones IPv6 a los routers y hosts. Como lo hace IPv4, los routers IPv6 recibirán la asignación de direcciones manualmente como parte del proceso de definición de la topología. La asignación de direcciones puede ser manual o tomar las ventajas de los mecanismos de asignación automática de direcciones definidos por IPv6.

3.2.1 ACTUALIZACIÓN DEL SISTEMA DE DOMINIO DE NOMBRES[142]

El DNS es usado tanto en IPv4 como en IPv6 para transformar los nombres de los hosts en direcciones IP. Un nuevo recurso llamado AAAA es definido para las direcciones IPv6. Los tipos en IPv4 se definen como A. Esto es así porque en IPv6 las direcciones son 4 veces más grandes de lo que son en IPv4. El DNS debe ser capaz de resolver las direcciones de los hosts ya sean de IPv4 o de IPv6.

Antes de que un servidor DNS pueda servir como un cliente doble capa IP, debe ser actualizado para manejar el nuevo tipo AAAA. Los servidores DNS que proveen soporte AAAA no necesitan ser actualizados para hacer uso del IPv6 para la transferencia de datos entre los servidores DNS.

Para sitios que no han sido actualizados sus DNS, los nodos IP pueden resolver direcciones de redes usando tablas de hosts definidas manualmente. Estos son archivos que residen en hosts que convierten los nombres en direcciones IP.

---

[141] RFC 1933 Transition Mechanisms for IPv6 Hosts and Routers
[142] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997.

3.2.2 APLICACIÓN[143]

Cuando las redes de doble capa IP sean actualizadas, las aplicaciones tradicionales de IPv4 pueden ser corridas sin ningún cambio. El Tunneling provee una manera de utilizar una infraestructura ya establecida de IPv4 para llevar el tráfico de IPv6 mientras haya recursos nativos de IPv6 que explotar.

Mientras la infraestructura de IPv6 se actualiza y entra en vigor, la infraestructura existente de IPv4 queda funcional y puede ser utilizada para llevar tráfico de IPv6 con routers que soporten topologías IPv4 e IPv6 los cuales encapsulan los paquetes de IPv6 en IPv4.

El tunneling aligera el sistema de encaminamiento de IPv6, si tenemos una red con IPv6 en parte de ella, el encaminamiento será más eficiente si se realiza el tunneling de IPv6, en lugar de cambiar de golpe toda la topología.

El tunneling ayuda a activar el servicio global de IPv6 durante la transición. Soporta una estrategia de transición mediante la cual la infraestructura del encaminamiento de IPv6 puede crecer con el paso del tiempo sin que tenga que cambiarse todo drásticamente. Es decir se puede hacer la transición de lo que se necesite, cuando se necesite.

Aunque existen muchos métodos de tunneling la mayoría de los utilizados son básicamente lo mismo. Para enviar un paquete al túnel, un nodo crea y encapsula el encabezado de IPv4 y lo transmite. La dirección destino del paquete encapsulado de IPv4 contiene la dirección del nodo donde termina el túnel, donde se va a recibir el encabezado del paquete encapsulado de IPv4, éste lo actualiza a un encabezado de IPv6 y luego procesa el paquete de IPv6 como si fuera cualquier otro paquete.

[143] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997.

3.3 ENRUTAMIENTO Y REDES CON CAPA IP DUAL

La transición de la capa IP dual permite que IPv4 e IPv6 sean enrutadas independientemente. Mucha de la flexibilidad de la estrategia de transición IPv6 proviene del hecho de que los enrutadores se ocupen ya de protocolos múltiples. Es común en redes actuales encontrar enrutadores que utilicen IPv4, IPX, DECnet, SNA, AppleTalk, y otros protocolos simultáneamente. Cada protocolo es utilizado generalmente por un protocolo separado de enrutamiento usando las estructuras independientes de la dirección. Por lo tanto, un protocolo adicional no es un cambio importante.

IPv6 puede utilizar el mismo protocolo de enrutamiento (RIP o OSPF) que IPv4 con lo que el trabajo de los enrutadores se puede simplificar en algo, además puede utilizar el mismo protocolo de manejo (SNMP), así como también el servicio de DNS. Con lo que se consigue un traslapo administrativo considerable con la infraestructura IPv4.

Algo no tan importante es el uso de un solo protocolo de enrutamiento integrado para soportar el enrutamiento tanto de IPv4 como para IPv6. Aún no hay protocolo que tenga esta característica, pero si se desarrolla tal protocolo, no cambiaría la naturaleza básica de la capa IP dual.

3.4 ESTRUCTURA DE REDES IP DUAL[144]

Con una estrategia transitiva pura de pila dual, las configuraciones del IPv6 y las redes IPv4 pueden ser desemparejadas lógicamente, a pesar de que están utilizando la misma infraestructura física. Las funciones existentes de IPv4 siguen siendo independientes y no afectadas, incluyendo los aspectos tales como la alimentación de la información de la ruta entre los dominios de enrutamiento y las asignaciones locales de direccionamiento. En efecto, la topología IPv6 se construye sin mucha profundidad.

---

[144] RFC 1933 Transition Mechanisms for IPv6 Hosts and Routers

Algunos diseñadores de redes piensan en la nueva configuración IPv6 como independiente, hay algunas ventajas al alinear las estructuras lógicas de las dos redes. En muchos casos IPv4 y los límites del dominio IPv6 pueden ser iguales, así que la espina dorsal y la estructura de la organización de la empresa pueden ser conservadas. Esto implica usar los mismos límites del dominio y de área para particionar la topología.

Mientras la nueva red IPv6 utilice una capa real de IP dual y direcciones reales IPv6, puede considerarse como una configuración independiente que puede desarrollarse en su propia trayectoria. Si por ejemplo, la dirección actual IPv4 no es ideal, porque los diseñadores no pueden utilizar direccionamiento basado en CIDR o si IPv4 tiene una estructura inconveniente del área, estos problemas se podrían fijar en una red independiente IPv6 con un esfuerzo mínimo.

3.5 DESCRIPCIÓN DE CONSTRUCCIÓN DE TUNELES[145]

La infraestructura de enrutamiento IPv6 será lanzada en un cierto plazo. La construcción de un túnel proporciona una manera de utilizar una infraestructura existente del enrutamiento IPv4 para controlar el tráfico IPv6 a pesar de que no hay recursos propios para IPv6. Mientras se desarrolla IPv6, la infraestructura existente del enrutamiento IPv4 puede seguir siendo funcional y se puede utilizar para controlar el tráfico de IPv6. Los servidores IPv6/IPv4 y los enrutadores hacen un túnel para los datagramas IPv6 sobre regiones con topología para enrutamiento IPv4 encapsulándolas dentro de los paquetes IPv4. La construcción de túneles puede simplificar el proceso de transición para los usuarios tan bien como proporcionar otro número de ventajas y le da fuerza al sistema existente del enrutamiento IPv4 para construir un sistema de enrutamiento IPv6. En casos donde la topología inicial IPv6 representa una porción pequeña de la red total, puede ser considerablemente más eficiente hacer un túnel IPv6 que construir totalmente una nueva topología IPv6.

[145] RFC 1933 Transition Mechanisms for IPv6 Hosts and Routers

La construcción de túneles ayuda a activar un servicio global IPv6 en la transición. Quienes adopten IPv6, que pueden estar geográficamente dispersos, pueden utilizar un túnel para proporcionar conectividad global con IPv6 sin esperar que Internet se cambie totalmente a IPv6. La construcción de túneles utiliza una estrategia de transición en la cual la infraestructura del enrutamiento IPv6 pueda crecer gradualmente y sea traída a sitios donde se necesita.

Aunque hay una variedad de métodos para la construcción de túneles, la mayoría de los mecanismos subyacentes son iguales. Para enviar un paquete en un túnel, un nodo primero crea un encabezado de encapsulamiento IPv4, y enseguida, transmite el paquete encapsulado.

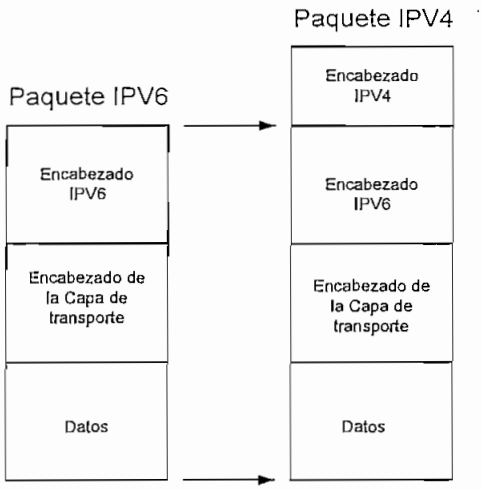


Fig. 3.1 Encapsulado IPv6 en IPv4[146]

La dirección destino del paquete de encapsulamiento IPv4 especifica el túnel para el nodo que recibe el paquete encapsulado extraído del encabezado de encapsulamiento IPv4, actualiza el encabezado IPv6, y después procesa el paquete incluido IPv6 como cualquier otro paquete recibido.

[146] RFC 1933 Transition Mechanisms for IPv6 Hosts and Routers

3.6 AUTOMATIZACIÓN CONTRA CONFIGURACIÓN DE TÚNELES[147]

Existen dos métodos importantes para la construcción de túneles: hacer un túnel configurado y hacer un túnel automático. El túnel configurado emplea métodos tradicionales, en los cuales las conexiones lógicas individuales del túnel se configuran entre dos nodos, por lo común enrutadores, que son separados por una topología arbitraria IPv4. Estas conexiones lógicas de tres capas son tratadas por nodos de tunelamiento como conexiones virtuales de punto a punto. Cada conexión del túnel es configurada manualmente asignando direcciones IP a una o a ambos extremos del túnel.

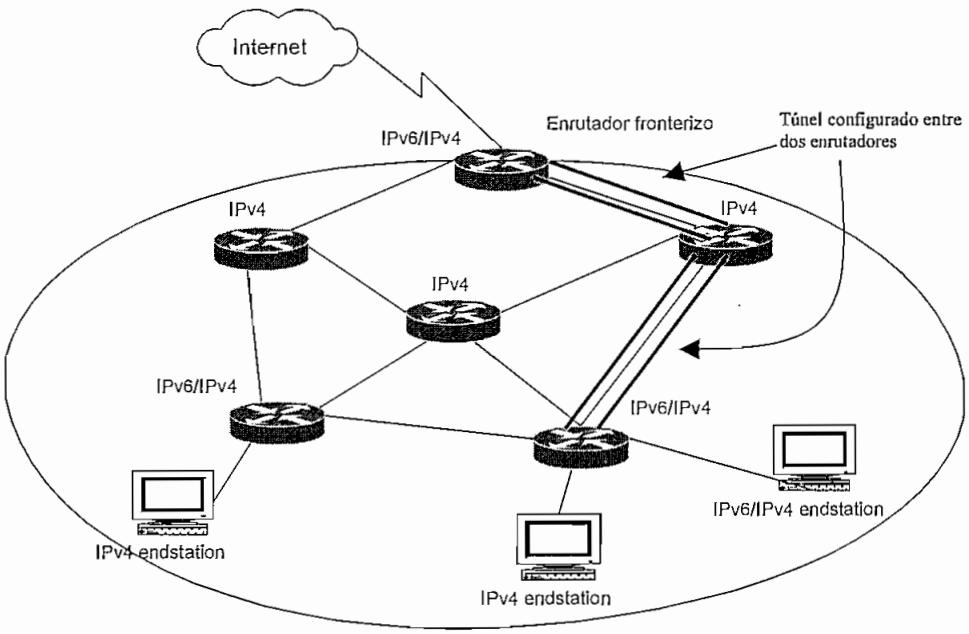


Fig. 3.2 Tunelamiento configurado[148]

La construcción de túneles automáticos utiliza los mismos mecanismos que el túnel configurado, pero elimina la necesidad de configurar cada túnel

[147] RFC 1933 Transition Mechanisms for IPv6 Hosts and Routers
[148] http://telecom.noc.udg.mx/~kenya/ipv6.html

individualmente. Se define un formato especial de direccionamiento IPv6, el cual es compatible con IPv4, el cual guarda una dirección IPv4 en los 32 bits de orden inferior. Una dirección compatible con IPv4 es identificada con un prefijo de 96 bits todos en 0, y mantiene una dirección IPv4 en los 32 bits de orden inferior.

0000.....000000000	*0000*	*Dirección IPv4*
(80 bits)	*(16 bits)*	*(32 bits)*

Fig. 3.3 Formato de dirección IPv6 compatible con IPv4[149]

Una dirección compatible con IPv4 puede ser vista como una dirección sencilla que sirve tanto para direcciones IPv6 como IPv4. Se utiliza la dirección IPv6 entera de 128 bits compatible con IPv4 como la dirección del nodo, mientras que la dirección IPv4 establecida en los 32 bits de orden inferior sirve como la dirección IPv4 del nodo. La dirección IPv4 establecida se asigna según el plan de direccionamiento IPv4.

Los nodos que tienen ya una asignación de dirección IPv4 pueden utilizar ese direccionamiento en una dirección compatible con IPv4. Los servidores que utilizan tunelamiento automático tienen asignadas direcciones IPv6 de esta forma. Cuando un nodo IPv6 desea entregar un paquete IPv6 que está direccionado en una dirección IPv6 compatible con IPv4, éste puede transmitirlo a través de un túnel de la estructura de enrutamiento IPv4 hasta su destino usando el direccionamiento IPv4 embutido en la dirección IPv6 de destino. Puesto que la dirección del extremo del túnel está implícita en la dirección destino compatible del paquete, esta forma de construir el túnel se utiliza solamente al enviar los paquetes a su destino final. Por lo que los túneles automáticos se utilizan típicamente para enviar paquetes a los servidores, no para las conexiones entre enrutadores.

[149] http://telecom.noc.udg.mx/~kenya/ipv6.html

Los nodos IPv6/IPv4 que realizan tunelamiento automático pueden utilizar protocolos para la configuración de direcciones IPv4 tales como DHCP, BOOTP o RARP para interpretar sus direcciones IPv6 compatible con IPv4. Esto lo hacen simplemente anteponiendo el prefijo 0:0:0:0:0:0 de 96 bits todos en cero a la dirección IPv4 que se realiza mediante el protocolo de configuración de IPv4. Este modo de configuración permite que los nodos IPv6/IPv4 fortalezcan la base ya instalada de los servidores de configuración de direcciones IPv4. Puede ser útil en los ambientes donde los enrutadores IPv6 y los servidores de configuración de direcciones aún no se hayan implementado.

3.7 TUNELAMIENTO APLICADO[150]

Debido a que los hosts y los enrutadores pueden desempeñar el papel de extremo de un túnel, hay un número de configuraciones factibles para hacer un túnel. Algunas de estas configuraciones se prestan para hacer un túnel automático mientras que otras requieren el método de tunelamiento que tiene la siguiente estructura:

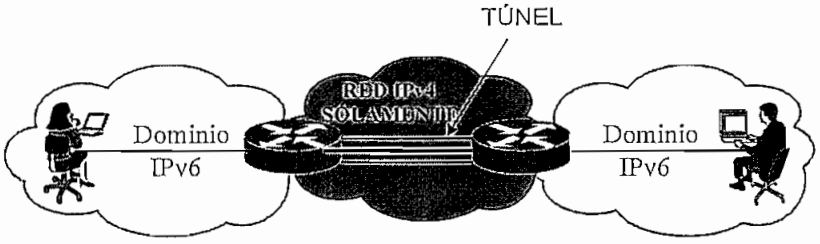


Fig. 3.4 Tunelamiento IPv6 sobre IPv4[151]

- **Enrutador a enrutador:** Los enrutadores IPv6/IPv4 interconectados por una infraestructura IPv4 pueden transmitir paquetes IPv6 por túneles entre sí mismos. El túnel atraviesa un segmento del camino punto a punto que el paquete IPv6 toma.

[150] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997
[151] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997

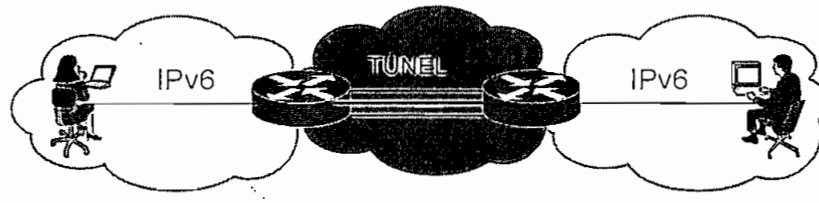


Fig. 3.5 Tunelamiento enrutador a enrutador[152]

- **Host a enrutador:** Los hosts IPv6/IPv4 pueden transmitir paquetes IPv6 por un túnel hacia un enrutador intermediario IPv6/IPv4 que sea accesible por medio de una infraestructura IPv4. Este tipo de túnel atraviesa el primer segmento de la trayectoria punto a punto.

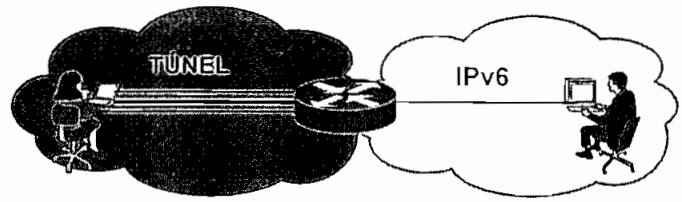


Fig. 3.6 Túnel Host a enrutador[153]

- **Host a Host:** Los hosts IPv6/IPv4 que están interconectados por una infraestructura IPv4 pueden transmitir paquetes IPv6 por un túnel entre sí mismos. El túnel atraviesa el camino entero punto a punto.

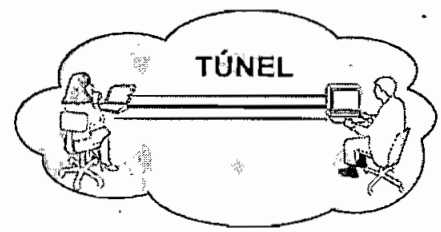


Fig. 3.7 Túnel host a host

- **Enrutador a Host:** Los enrutadores IPv6/IPv4 pueden transmitir paquetes IPv6 con su host destino final IPv6/IPv4. Este túnel atraviesa solamente el último segmento de la trayectoria punto a punto.

[152] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997
[153] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997

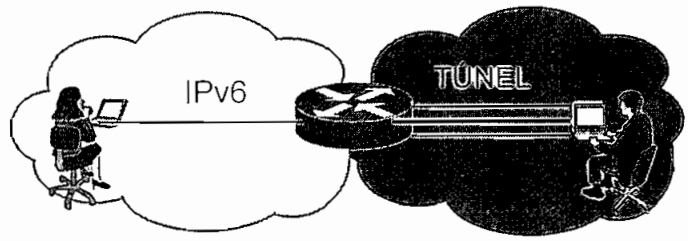


Fig. 3.8 Túnel enrutador a host[154]

En los dos primeros métodos de tunelamiento, enrutador a enrutador y host a enrutador, el paquete IPv6 es tunelado a un enrutador. Estas configuraciones por lo común utilizan la construcción de túneles configurados. El extremo del túnel es un enrutador intermedio el cual debe desencapsular el paquete IPv6 y redirigirlo a su destino final. Cuando se hace un túnel a un enrutador, el extremo de un túnel es diferente de su último destino así que las direcciones en los paquetes IPv6 al ser tunelados no proporcionan la dirección IPv4 del extremo del túnel en lugar de esto la dirección del extremo del túnel debe ser determinado por la información de configuración en el nodo que realiza el tunelamiento.

En cada túnel configurado, el nodo debe salvar la dirección del extremo del túnel. Cuando un paquete IPv6 se transmite sobre un túnel, la dirección del extremo del túnel se utiliza como la dirección destino para el encabezado de encapsulado IPv4.

3.8 TÚNEL CONFIGURADO POR DEFECTO[155]

IPv6 o los nodos duales que están conectados con las infraestructuras de enrutamiento IPv4 pueden utilizar un túnel configurado para obtener un backbone IPv6. Si la dirección IPv4 de un enrutador IPv6/IPv4 es conocida, un túnel se puede configurar a ese enrutado en el vector de enrutamiento como una *ruta por defecto*. Es decir, todas las direcciones destino IPv6 coincidirán con la ruta y podrán atravesar el túnel. Puesto que la longitud de la máscara de la ruta por

[154] Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997
[155] RFC 1933 Transition Mechanisms for IPv6 Hosts and Routers

predefinición es cero, será utilizada solamente sino no hay otras rutas con una máscara más larga que correspondan con el destino.

La dirección del extremo del túnel por defecto podrá ser la dirección IPv4 de un enrutador IPv6/IPv4 en la frontera del backbone IPv6. Como alternativa el extremo del túnel podrá ser una dirección IPv4 de cualquier tipo de transmisión. Con este acercamiento, los enrutadores múltiples IPv6/IPv4 fronterizos informan sobre la escalabilidad o alcance de IPv4 en la misma dirección IPv4. Todos estos enrutadores aceptan los paquetes de esta dirección como propios, y desencapsularán los paquetes IPv6 tunelados a esta dirección. Cuando un nodo IPv6/IPv4 envía un paquete encapsulado a esta dirección, será entregado solamente a uno de los enrutadores fronterizos, pero el nodo que envía no sabrá a cual de todos. El sistema de enrutamiento IPv4 transportará generalmente el tráfico al enrutador más cercano.

Utilizar un túnel de valor por defecto a una dirección IPv4 de cualquier tipo de transmisión proporciona un alto grado de robustez puesto que los enrutadores múltiples fronterizos pueden ser utilizados y, con los mecanismos normales de retraso de enrutamiento IPv4, el tráfico cambiará automáticamente a otro enrutador cuando uno de ellos se cae.

3.9 TUNELAMIENTO AUTOMÁTICO[156]

Cuando un host es el extremo de un túnel (en el tunelamiento host a host y enrutador a host), los paquetes IPv6 son tunelados en todo su camino hasta su destino final.

Estas configuraciones por lo general usan tunelamiento automático. En este caso, el extremo del túnel es el último destino de un paquete IPv6, el extremo del túnel puede ser determinado por la dirección IPv6 destino del paquete: si la dirección es compatible con IPv4 entonces los 32 bits menos significativos guardan la

[156] http://telecom.noc.udg.mx/~kenya/ipv6.html

dirección IPv4 del nodo destino, y eso puede ser usado como la dirección destino del extremo del túnel. Por lo tanto, el tunelamiento automático elimina la necesidad de configurar explícitamente la configuración del extremo del túnel. Los paquetes IPv6 que no son direccionados a una dirección compatible con IPv4 no pueden usar tunelamiento automático.

Los nodos IPv6/IPv4 necesitan determinar cuales paquetes IPv6 pueden ser enviados por medio del tunelamiento automático. Una técnica es usar la tabla de enrutamiento IPv6 para dirigir el tunelamiento automático. La implementación puede tener un registro en una tabla especial fija de enrutamiento para el prefijo 0:0:0:0:0:0/96 (esto es, una ruta al prefijo con puros ceros con una máscara de 96 bits). Los paquetes que concuerden con este prefijo son enviados a un controlador de pseudo-interface que realiza el tunelamiento automático. A partir de que todas las direcciones IPv6 compatibles con IPv4 concordarán con este prefijo, todos los paquetes dirigidos a esos destinos serán auto-tunelados (a menos que exista una mejor ruta de concordancia).

3.10 TUNELAMIENTO y DNS[157]

Cuando una dirección IPv6 compatible con IPv4 es asignada a un host IPv6/IPv4 que soporta tunelamiento automático, el registro A y los registros AAAA correspondientes pueden ser listados en el DNS. Los registros AAAA guardan la dirección IPv6 compatible con IPv4 mientras que el registro A guarda los 32 bits menos significativos de esa dirección. El registro AAAA será ubicado por solicitudes de hosts IPv6 mientras que el registro A será encontrado por solicitudes de hosts IPv4 únicamente.

La decisión de establecer un registro AAAA que guarde una dirección compatible con IPv4 en el DNS para un host IPv6 puede ser utilizada como una política de control para el tráfico hacia ese host. Si una dirección compatible con IPv4 es

---

[157] http://telecom.noc.udg.mx/~kenya/ipv6.html

listada entonces otros hosts originarán tráfico tunelado a ese host. Si solo un registro A es listado entonces el host "aparecerá" para los otros como un host IPv4 únicamente y solo tráfico IPv4 será enviado.

Cuando una petición de un host IPv6/IPv4 localiza un registro AAAA que guarda una dirección IPv6 compatible con IPv4, igual de efectivo como cuando un registro A guarda la dirección con IPv4 correspondiente, las librerías de resolución no requieren necesariamente de devolver las dos direcciones a la aplicación. Tiene tres opciones:

- Devolver solo la dirección IPv6 a la aplicación
- Devolver solo la dirección IPv4 a la aplicación
- Devolver las dos direcciones a la aplicación

La determinación de cuáles direcciones devolver puede ser utilizada como una política de selección en el host para controlar el tráfico originado por ese host. Si el administrador del sistema desea prevenir que su host origine tráfico tunelado, puede configurar las librerías de resolución para devolver solo direcciones IPv4 a la aplicación. Si se permite el tráfico tunelado entonces el administrador permite que direcciones IPv6 (compatibles con IPv4) sean devueltas a la aplicación.

3.11 CARACTERÍSTICAS DE LA IMPLEMENTACIÓN DEL TUNELAMIENTO[158]

Los nodos que realizan tunelamiento necesitan ajustarse a una serie de características que son comunes tanto para el tunelamiento automático y configurado. Muchas de estas características tratan el hecho de que la topología y operaciones de enrutamiento de un túnel IPv4 es demasiado transparente para los nodos IPv6 que la usan.

[158] RFC 1933 Transition Mechanisms for IPv6 Hosts and Routers

- ***Túnel MTU y fragmentación***. Es técnicamente factible para un nodo de encapsulamiento manejar a un túnel como una conexión virtual IPv6 con un MTU largo, dependiendo de una capa IPv4 de fragmentación y de un reensamble para liberar paquetes IPv6 que son más grandes que el MTU de las subyacentes conexiones de la ruta entre el nodo de encapsulamiento y de desencapsulamiento. Pero esto nos lleva a una fragmentación interna del túnel lo cual es ineficiente. Una mejor perspectiva es hacer que el nodo de encapsulamiento interprete rutas IPv4 del MTU de la ruta del túnel, y después usar ambos, la ruta IPv6 para reportar al MTU de regreso al host origen.

- ***Mantenimiento del estado de información del estado***. Si el nodo de encapsulamiento interpreta rutas IPv4 del MTU de sus túneles, necesitará mantener un estado de información para cada túnel. A partir de que el número de túneles que un nodo puede estar usando puede crecer gradualmente, este nodo debe emplear un esquema de refrescamiento del estado de información que necesita, y periódicamente descartar la información que no esté siendo utilizada.

- ***Límite de salto del IPv6 e IPv4 TTL***. Los túneles IPv6 en IPv4 son tratados como conexiones de salto sencillo desde el punto de vista de IPv6. Esto es, que el límite de salto de IPv6 es decrementado por uno cuando un paquete IPv6 atraviesa un túnel. Pero el nodo de encapsulado debe utilizar un valor de la TTL en el encabezado del encapsulado IPv4 que es bastante grande como para garantizar que el paquete encapsulado no expirará en el túnel, ruteado al nodo de desencapsulamiento.

- ***Mensajes de error de IPv4 ICMP y tunelamiento.*** Los paquetes tunelados pueden fallar al ser liberados al extremo del túnel porque un extremo puede ser incomunicable, el TTL de IPv4 no es lo suficientemente grande o el paquete es demasiado grande. Estas fallas mandarán mensajes de IPv4 de ICMP de error dirigidos de vuelta al punto de entrada del túnel, de enrutadores IPv4 a lo largo del camino del túnel. Algunos de

estos mensajes de error del ICMP pueden contener bastante del paquete original IPv6 para identificar su origen, puesto que muchos enrutadores IPv4 devuelven solamente 8 bytes de datos más allá que el encabezado IPv4 del paquete de error. Si es posible, el nodo de encapsulado debe procurar recuperar el paquete original IPv6, y generar el mensaje de error apropiado del ICMP IPv6 de nuevo al nodo origen.

3.12 TUNELAMIENTO Y ENRUTAMIENTO DE PROFUNDIDAD[159]

Durante un período extendido de transición de IPv4 a IPv6, las infraestructuras de enrutamiento de IPv4 e IPv6 estarán presentes. En la operación pura básica de la capa de IP dual, el enrutamiento puede ser independiente de cualquiera de estas dos infraestructuras. Sin embargo, inicialmente, los dominios IPv6 podrían no estar globalmente interconectados en una infraestructura Internet IPv6 y por lo tanto puede necesitar comunicarse usando tunelamiento a través de redes con backbone de IPv4 únicamente.

Para lograr el enrutamiento dinámico en un ambiente tan híbrido, se necesitan mecanismos globales para distribuir la capa de red IPv6 a los nodos de enrutamiento en los dominios IPv6 dispersos. (Alternativamente, algunas de las mismas técnicas se pueden utilizar en fases posteriores de la transición al ruteo IPv4 de paquetes entre redes aisladas IPv4 únicamente sobre backbone IPv6).

El uso de tunelamiento requiere el estado coherente de rutas entre IPv4 e IPv6. Por ejemplo, considere un paquete que comience como paquete IPv6, pero después pasa a través de un túnel IPv4 (es decir, se encapsula en un paquete IPv4) en el centro de su camino desde el origen a su destino. Este paquete se debe enrutar (con un enrutamiento IPv6) al extremo inicial correcto del túnel, atravesar el túnel como un paquete IPv4, y después atravesar el resto del camino otra vez como paquete IPv6. Este paquete tiene que seguir claramente una ruta constante en todo el camino desde el origen hasta el destino. Las implicaciones

[159] http://telecom.noc.udg.mx/~kenya/ipv6.html

de éste proceso en el enrutamiento se discuten por separado para los túneles de enrutador a enrutador, los túneles del host a host, los túneles del host a enrutador, y los túneles del enrutador a host.

3.13 TÚNELES DE ENRUTADOR A ENRUTADOR[160]

Los túneles de enrutador a enrutador se basan en la configuración manual de ambos extremos del túnel. Específicamente, el enrutador en cada extremo del túnel se debe configurar manualmente para saber las direcciones asociadas al final de la conexión. Tales túneles también se refieren como túneles "completamente manualmente configurados", puesto que ambos finales de la conexión deben ser configurados.

En los túneles de enrutador a enrutador que maneja IPv6 sobre IPv4, los enrutadores tratan la conexión como si fuera una conexión normal punto a punto. Por ejemplo los protocolos dinámicos de enrutamiento tales como OSPF o BGP/IDRP pueden enviar la información de escalabilidad a través de esta conexión al igual que en cualquier otro tipo de conexión. La decisión para remitir un paquete a través de un túnel enrutador a enrutador configurado manualmente se hace de manera semejante a remitir un paquete a través de cualquier otro tipo de conexión. Específicamente, los paquetes se remiten basados en las rutas computadas por protocolos estándares de enrutamiento. Estas rutas pueden utilizar conexiones normales y conexiones tuneladas en cualquier combinación.

El uso de los túneles enrutador a enrutador manualmente configurados tiene la ventaja de que la infraestructura subyacente es transparente a los protocolos que se remiten a través del túnel. Por ejemplo, si es IPv6 tunelado a través de IPv4, después la infraestructura IPv4 se utiliza para la expedición IPv6, pero los detalles internos de la infraestructura IPv4 no importan a los enrutadores IPv6 ni a los protocolos de enrutamiento IPv6. También, todos los tipos de direcciones IPv6 sin excepción pueden ser anunciados en el enrutamiento IPv6 y tunelado a través de las redes IPv4.

Puesto que se encapsulan los paquetes IPv6 solamente cuando viajan a través de los segmentos de la red que no utilizan IPv6, y se remiten según sus encabezados nativos a otra parte, este método no obliga a los tipos de políticas de enrutamiento las cuales pueden emplear a través de la porción IPv6 del camino de datos.

Sin embargo, un túnel enrutador a enrutador manualmente configurado difiere de una conexión normal en un aspecto importante: en muchos casos es probable tener un funcionamiento más bajo, tal como rendimiento de procesamiento más bajo o mayor retardo. El uso de un protocolo de enrutamiento tal como RIP, que trata cada conexión por igual, podría conducir las rutas subóptimas. Sin embargo, esto no es un problema con protocolos más flexibles de enrutamiento tales como OSPF, que permite un rango amplio dinámico en la métrica asignada a cada conexión, específicamente con el OSPF, la conexión tunelada podía ser dada gracias a un costo más alto comparado a otras conexiones.

En los túneles enrutador a enrutador se configuran manualmente ambos extremos de la conexión y en un principio, los túneles host a enrutador o host a host se pueden configurar de la misma manera. Sin embargo, cuando un número de hosts están implicados como extremos, el requisito de que cada extremo del túnel sea configurado hace que los túneles "completamente manuales" sean menos útiles.

3.14 EL TUNELAMIENTO AUTOMÁTICO DE HOST A HOST

Si el origen y los hosts destino hacen uso de direcciones IPv6 compatible con IPv4, entonces es posible que el tunelamiento automático sea utilizado para la trayectoria entera del host de origen al host destino. En este caso, el paquete IPv6 es encapsulado en un paquete IPv4 por el host de origen, y remitido por los

[160] RFC 1933 Transition Mechanisms for IPv6 Hosts and Routers

enrutadores como un paquete IPv4 todo el camino hasta el host destino. Según lo discutido anteriormente, esta característica permite la implementación inicial de los hosts IPv6 antes de que cualquier enrutador sea actualizado.

Un host de origen puede hacer uso de tunelamiento automático de host a host a condición de que lo siguiente sea verdad.

- La dirección de origen es una dirección IPv6 compatible con IPv4.
- La dirección destino es una dirección IPv6 compatible con IPv4
- El host de origen no tiene conocimiento de ningún enrutador IPv6 vecino
- El host de origen tiene conocimiento de uno o más enrutadores IPv4 vecinos.

Si todos estos requisitos se cumplen entonces el host de origen pueden encapsular el paquete IPv6 en un paquete IPv4, usando una dirección IPv4 de origen extraído de la dirección IPv6 asociada de origen, y una dirección IPv4 destino extraída de la dirección IPv6 asociada destino. Donde se utilice tunelamiento automático de host a host, el paquete se remite como paquete normal IPv4 para todo su camino, y es desencapsulado (es decir, se quita la encabezado IPv4) solamente por el host destino. El tunelamiento automático de host a host requiere que el enrutamiento normal IPv4 sea operacional, pero no impone requisito en el enrutamiento IPv6.

3.15 TÚNELES DE HOST A ENRUTADOR[161]

En algunos casos un host de la capa de IP dual puede llegar a necesitar transmitir un paquete IPv6, pero puede ser que no tenga un enrutador IPv6 local que pueda utilizar para este propósito. En lugar de esto, el host puede utilizar tunelamiento a un enrutador IPv6. Esta capacidad permite que el host transmita los paquetes con un número arbitrario de nodos IPv4 a un backbone IPv6, que entonces alternadamente transmitirá los paquetes usando la expedición normal IPv6. Los túneles host a enrutador pueden ser logrados configurando manualmente el host

[161] RFC 1933 Transition Mechanisms for IPv6 Hosts and Routers

de la capa de IP dual con una dirección IPv4 que pueda utilizar para llegar hasta el backbone IPv6.

Para que la conversación trabaje en ambas direcciones es necesario que un túnel enrutador a host tenga un camino de vuelta. Esto requiere que cualquiera de ambos extremos del túnel se configuren manualmente, o que se utilice el tunelamiento automático en la dirección posterior (del enrutador a host). Este último tipo de túnel se puede referir como "túnel semimanualmente configurado", puesto que la configuración manual se utiliza en una dirección, pero el tunelamiento automático se utiliza en la otra.

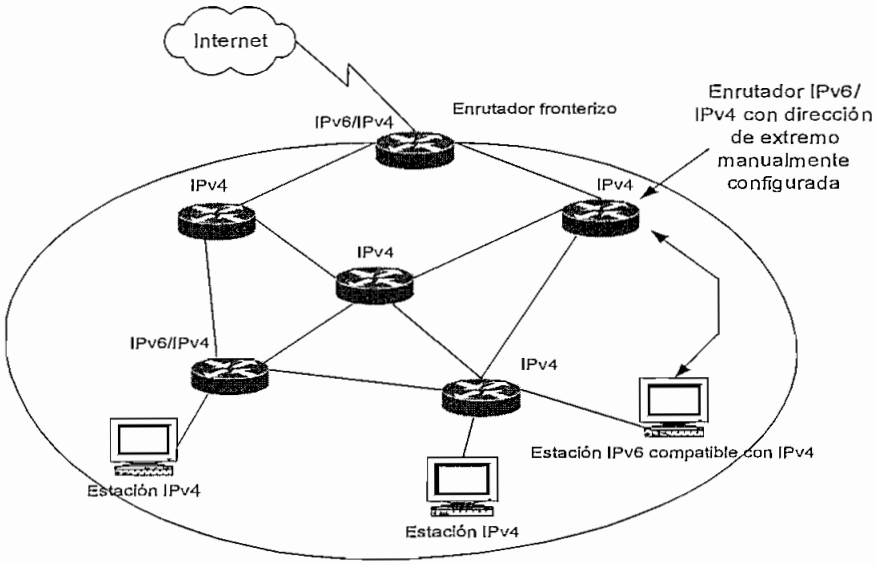


Fig. 3.9 Túnel semimanual[162]

Un túnel semimanualmente configurado ocurre cuando el host es configurado para que pueda encontrar al enrutador pero el enrutador no se configura con ningún conocimiento específico del host y tunelamiento automático para encontrarlo. Esto, por supuesto, requiere que el host tenga una dirección IPv6 compatible con IPv4 y que el host esté configurado con una dirección IPv4 para utilizarla como túnel hacia el enrutador IPv6.

[162] RFC 1933 Transition Mechanisms for IPv6 Hosts and Routers

Un host de origen puede hacer uso de tunelamiento semimanualmente configurado de host a enrutador a condición de que se cumpla lo siguiente:

- La dirección de origen es una dirección IPv6 compatible con IPv4
- El host de origen no tiene conocimiento alguno de ningún enrutador IPv6 vecino
- El host de origen no tiene conocimiento alguno de ningún enrutador IPv4 vecino
- El host de origen es configurado con una dirección IPv4 de un enrutador, que puede hacer como extremo del túnel.
- La dirección destino no es compatible con IPv4 (sí la dirección destino es compatible con IPv4 entonces se utilizará el tunelamiento automático).

Si todos estos requisitos se cumplen entonces el host de origen puede encapsular el paquete IPv6 en un paquete IPv4, usando una dirección IPv4 de origen que se extraiga de la dirección IPv6 asociada al origen, y una dirección IP destino que corresponda a la dirección configurada del enrutador dual que sirve como extremo del túnel.

3.16 ESCALABILIDAD PARA LOS TÚNELES DE HOST A ENRUTADOR[163]

El enrutador dual, que sirve como extremo del túnel semimanual debe avisar su disponibilidad dentro del enrutamiento IPv4 con el objetivo de provocar que el paquete encapsulado le sea remitido. El enfoque más simple es que una sola dirección IPv4 sea asignada al enrutador para el uso como extremo del túnel. Un enrutador dual de tunelamiento con conectividad a un backbone IPv6 puede avisar una ruta del host a esta dirección (en la red IPv4 solamente). Cada host dual en la red asociada de IPv4 únicamente se configura con la dirección de este extremo del túnel.

[163] RFC 1933 Transition Mechanisms for IPv6 Hosts and Routers

3.17 EL TUNELAMIENTO AUTOMÁTICO DE ENRUTADOR A HOST[164]

Si el tunelamiento se utiliza de un host a un enrutador del backbone IPv6, es también necesario utilizar tunelamiento del enrutador al host. En este caso (a condición de que el host destino tenga una dirección IPv6 compatible con IPv4) la expedición normal IPv6 se puede utilizar en parte del camino del paquete, y el tunelamiento automático de enrutador a host se puede utilizar para conseguir que el paquete de un enrutador dual sea encapsulado al host destino.

La expedición normal del paquete es directa en este caso: el enrutador de encapsulamiento crea el encabezado de encapsulado IPv4 usando una dirección IPv4 asignada a sí mismo como la dirección IPv4 de origen, y con una dirección IPv4 destino extraída de la dirección IPv6 destino compatible con IPv4. El paquete encapsulado se remite del enrutador de encapsulamiento al host destino usando el enrutamiento normal IPv4, en este caso la parte contrincante es el enrutamiento IPv6 requerida para entregar el paquete IPv6 del host de origen al enrutador de encapsulamiento. Para que esto suceda, el enrutador de encapsulamiento tiene que avisar su disposición para las direcciones IPv6 apropiadas compatible con IPv4 en la red IPv6.

El tunelamiento de enrutador a host por lo general ocurre cuando uno o más enrutadores de la capa de IP dual se ubican en el límite entre una red IPv4 únicamente y una red de la capa de IP dual. En este caso éstos "enrutadores fronterizos" necesitan avisar en el enrutamiento IPv6 (en la red dual) que ellos pueden interpretar ciertas direcciones IPv6 compatible con IPv4 que corresponden a las direcciones que existen en la red IPv4. En general esto requiere la configuración manual de los enrutadores fronterizos. Sin embargo, en la mayoría de los casos puede requerir que solamente uno o un número pequeño de prefijos de dirección sean avisados por toda la red local IPv4. Esto, por lo

[164] RFC 1933 Transition Mechanisms for IPv6 Hosts and Routers

CAPITULO 4

REQUERIMIENTOS PARA LA MIGRACIÓN DE IPv4 A IPv6 PARA UN DISTRIBUIDOR DE INTERNET

CAPÍTULO IV

REQUERIMIENTOS PARA LA MIGRACIÓN DE IPv4 A IPv6 PARA UN DISTRIBUIDOR DE INTERNET

El objetivo de este capítulo es estudiar los requerimientos que necesita un Distribuidor de Internet, que actualmente se encuentra funcionando con el Protocolo IP versión 4 para migrar al nuevo protocolo IP versión 6.

Inicialmente se describirá la red del Distribuidor de Internet, su funcionamiento con el protocolo IPv4 y, luego se analizará los cambios necesarios que se debe realizar en la misma para actualizarla con el nuevo protocolo IPv6.

4.1 DESCRIPCIÓN DE FUNCIONAMIENTO DEL BACKBONE DEL DISTRIBUIDOR DE INTERNET

Para analizar la migración del distribuidor de Internet de IPv4 a IPv6, se ha escogido una red privada que da servicio de Internet e incluye varios equipos de diferentes fabricantes. Un diagrama típico se muestra en la figura 4.1.

El backbone de un distribuidor de Internet está formado básicamente por los siguientes elementos:

- Router Principal: El cual constituye el corazón mismo del backbone, ya que a través de éste se enrutan los requerimientos hacia la Internet.

- Servidor de Acceso: Es el que permite el acceso de los clientes masivos hacia la Internet, éste reenvía su requerimiento al Servidor de Autenticación.

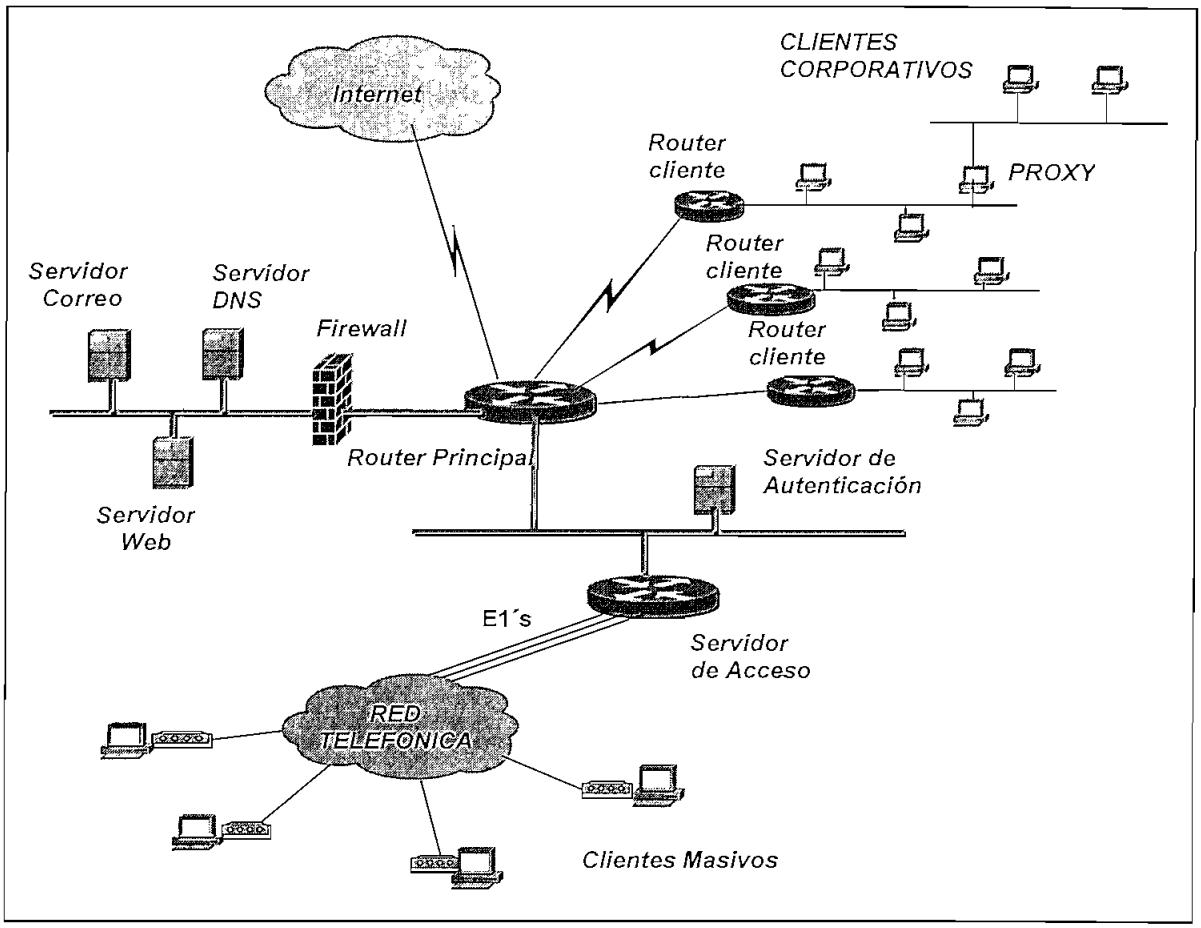


Fig. 4.1 Diagrama de red de un Distribuidor de Internet[165]

- *Servidor de Autenticación*: Es el que dependiendo del perfil de usuario acepta o niega el acceso; si el acceso es aceptado, éste a su vez envía los atributos correspondientes al servidor de acceso.

- *Firewall*: Este hace cumplir las políticas de seguridad para la comunicación entre las redes internas y externas, por ejemplo entre la red de los clientes y la red de servidores.

- *Red de Servidores*: Se denomina así a la red en donde se encuentran ubicados los diferentes servidores que dan servicio a los usuarios, éstos son por ejemplo, Servidor de Correo electrónico, Servidor de DNS, Servidor de Páginas WEB, etc.

Un distribuidor de Internet tiene básicamente 2 tipos de clientes que son: masivos y corporativos.

Los **clientes masivos** son aquellos que tienen solamente un computador y un módem y, el acceso es a través de la red pública telefónica, como se muestra en la figura 4.2. Estos hacen un requerimiento de autenticación al servidor de acceso con un nombre de usuario y un password, el servidor de acceso a su vez reenvía este requerimiento al servidor de autenticación, el cual responde aceptando o negando el acceso.

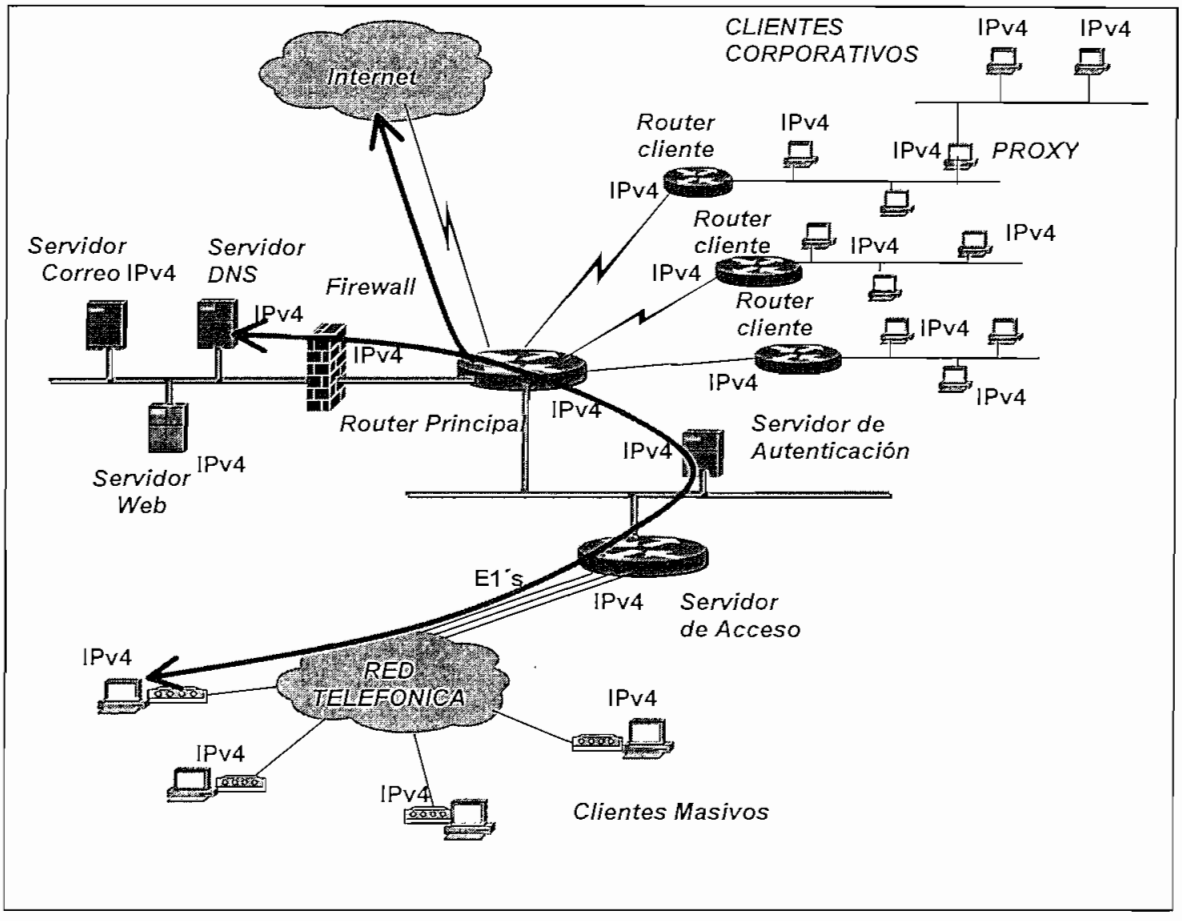


Fig. 4.2 Red operativa con el protocolo IPv4

[165] Proporcionado por la empresa de Telecomunicaciones IMPSATEL DEL ECUADOR

Todo host que está conectado a la Internet o que se va a conectar, necesita tener configurado un servidor de DNS, el cual se encargará de realizar la traducción de nombre de dominios a su dirección IP correspondiente en las aplicaciones que así lo requieran.

Los **clientes corporativos** son aquellos que tienen varios computadores interconectados a través de una red LAN. Este tipo de clientes no necesitan de un nombre de usuario y un password, ya que el acceso a la Internet se lo realiza a través de un router que se encuentra enlazado directamente con el router principal. De igual manera si un computador requiere conectarse a un sitio específico en la Internet y lo hace a través de un nombre de dominio, éste primero debe ser transformado a una dirección IP por el servidor de DNS.

Tanto para los clientes masivos como para los clientes corporativos, una vez que el host se conecta hacia el router principal, el acceso a la Internet se realiza de igual manera independientemente del tipo de cliente, es decir, la única diferencia entre los host de clientes corporativos y clientes masivos es el acceso hacia el router principal.

Todos los equipos involucrados en la red operan actualmente con el Protocolo IPv4, es decir los routers, servidores, hosts y el firewall utilizan direcciones de 32 bits de extensión para realizar sus requerimientos a la Internet.

4.2 ANÁLISIS DE LA RED OPERATIVA CON EL PROTOCOLO IPv4

Se analiza una red típica de un distribuidor de Internet, que opera con varias marcas de equipos que son los más comunes. El análisis se lo realizará de la siguiente manera:

1. Red con routers marca CISCO, servidores y firewall con sistema operativo Windows NT.

2. Red con routers marca CISCO, servidores y firewall con sistema operativo Sun Solaris.

En ambos casos se asume que los hosts de todos los clientes operan bajo el sistema operativo Windows NT.

El diseño de la red permite manejar como máximo 50 clientes corporativos y 500 clientes masivos.

4.2.1 ANÁLISIS PARA EL PRIMER CASO

En la figura 4.3 indica la red con los equipos que funciona actualmente.

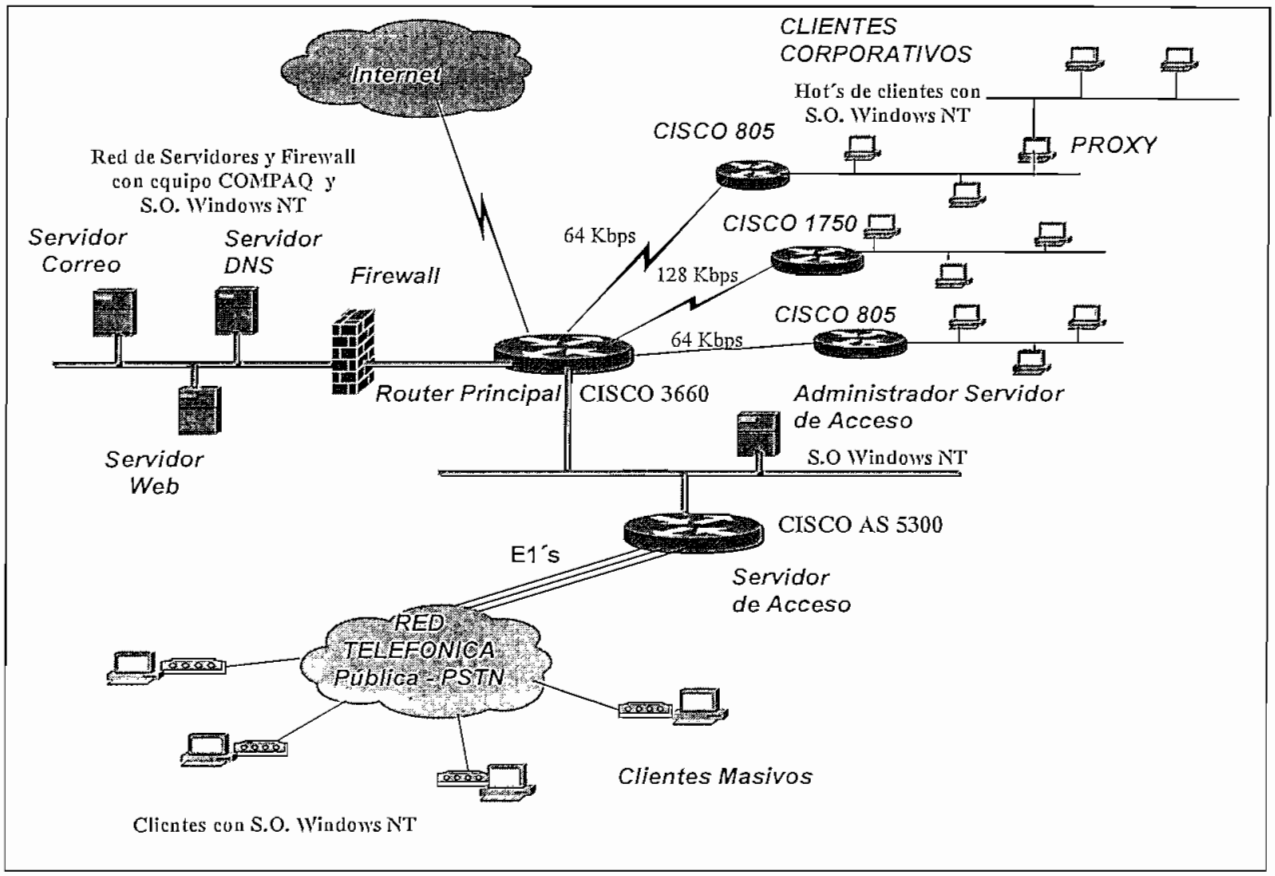


Fig. 4.3 Red operativa actual para el primer caso

Se analizan las características de cada equipo del backbone:

- **CISCO AS 5300:** Es un servidor de acceso universal y es el que permite el acceso remoto dialup de los clientes masivos hacia la Internet. Sus características son:

  - Soporta interfaces de red Ethernet, T1/PRI o E1/PRI, Serial sincrónico
  - Conexiones asincrónicas máximas 500
  - Memoria Flash por defecto 16 MB expandible a 32MB
  - Memoria RAM por defecto 64 MB expandible a 256 MB
  - Opciones de software IP/IP PLUS
  - Release IOS 12.0
  - Procesador de 150 MHz

- **CISCO 3660:** Es el router principal y, enruta todos los requerimientos tanto de clientes corporativos como masivos hacia la Internet. Sus características principales son:

  - Soporta interfaces de red Ethernet, Fast Ethernet, Token Ring, Asincrónicos, serial Sincrónico, Interfaz serial de alta velocidad, ISDN BRI, T1/ISDN PRI o E1/ISDN PRI
  - Memoria flash por defecto 8 MB expandible a 32MB
  - Memoria RAM por defecto 32 MB expandible a 256MB
  - Procesador de 80 MHz IDT
  - Opciones de software IP/IP PLUS
  - Release IOS 12.0

Las aplicaciones de los servidores de Correo, DNS, páginas WEB, autenticación y del firewall operan bajo el sistema operativo Windows NT. Para que este sistema funcione adecuadamente es necesario ciertos requerimientos mínimos de memoria como son:

- Memoria en RAM 32 MB
- Espacio libre en disco duro 110 MB
- Procesador Pentium

Las máquinas en las cuales se encuentran operando los servidores y el firewall son de marca COMPAQ que tienen una memoria en RAM de 64 MB y un disco duro de 1 GB.

Se considera el equipamiento que se encuentra en los clientes corporativos, ya que en los masivos no existen equipos instalados. Los clientes corporativos tienen instalado un router marca CISCO, para este análisis se consideran las velocidades de acceso de 64 Kbps y 256 Kbps por ser un promedio a nivel comercial en el Ecuador. Para los clientes que tienen contratado un acceso a 64 Kbps se les ha asignado el router CISCO 805, y para el acceso a 256 Kbps un router CISCO 1720. Sus características son:

- **CISCO 805:** Es el router que está conectado directamente a la red LAN del cliente y que a través de una última milla se une con el router principal del backbone. Sus características principales son:

  - 1 puerto serial y 1 puerto Ethernet 10BaseT
  - Memoria flash por defecto 4 MB expandible a 12 MB
  - Memoria DRAM por defecto 8 MB expandible a 16 MB
  - Procesador MPC 850 Velocidad 33 MHz
  - Release IOS 12.0

- **CISCO 1720:** Tiene la misma función que el router anterior. Sus características principales son:

  - 1 puerto LAN Fast Ethernet 10/100 BaseT y 2 puertos seriales
  - Memoria flash por defecto 8 MB expandible a 16 MB
  - Memoria DRAM por defecto 16 MB expandible a 32 MB
  - Procesador MPC 860T a 48 MHz

- Release IOS 12.0

4.2.2 ANÁLISIS PARA EL SEGUNDO CASO

La única variante son los servidores y el firewall que ahora funcionan bajo la plataforma de Sun Solaris. Los equipos en los que operan son los Sun Enterprise 220R (los más utilizados por los ISP´s) con el software Solaris 2.6.

Los requerimientos mínimos para ésta plataforma son:

- 48 MB de memoria en RAM
- 1.05 GB de espacio en disco duro

El equipo Sun Enterprise 220R tiene las siguientes características:

- Disco duro de 2 GB
- 64 MB de memoria en RAM.

4.3 ANÁLISIS DE LOS REQUERIMIENTOS PARA LA MIGRACIÓN A IPv6

Obtenidos los datos con los que opera la red, para los dos casos mencionados anteriormente, podemos analizar los requerimientos para la migración a IPv6. Se empezará por los equipos del backbone del distribuidor de Internet.

1. El servidor de acceso es de marca CISCO AS 5300 y opera actualmente con el release 12.0, para poder migrar a IPv6 es necesario realizar una actualización del software al release 12.2 cuyos requerimientos mínimos de memoria son:

   - 16 MB en memoria flash y 64MB en memoria RAM

El equipo tiene estos requerimientos mínimos, por lo que no será necesario realizar un upgrade de memoria.

2. El router principal de marca CISCO 3660, al igual que el anterior opera con el release 12.0, por lo que necesita actualizarse al release 12.2, para ello las necesidades mínimas de memoria son:

   - 8 MB en memoria flash y 32 MB en memoria RAM

   De igual manera este equipo tiene disponible los requisitos mínimos para operar con el nuevo release.

3. Los servidores y el firewall que funcionan con el sistema operativo Windows NT, necesitan realizar una actualización a otro sistema operativo como es Windows 2000, y cuyos requerimientos son:

   - 64MB de memoria en RAM y 650 MB de espacio libre en disco duro

   Con esto vemos que no existe inconveniente en memoria y por lo tanto solo requiere la actualización al nuevo sistema operativo.

4. Para los clientes corporativos se tienen 2 tipos de routers, el CISCO 805 y el CISCO 1720, ambos funcionan con el release 12.0 y deben actualizar al 12.2 de manera que puedan operar con el nuevo protocolo IPv6. Los requerimientos mínimos para cada router son:

   - CISCO 805: 4 MB en memoria flash y 8 MB en memoria RAM
   - CISCO 1720: 4 MB en memoria flash y 24 MB en memoria RAM

   Los routers tienen ésta característica y sólo necesitan actualizar su software.

5. En general los hosts de todos los clientes se asume que operan bajo la plataforma Windows NT, y deben realizar una actualización de su sistema operativo al nuevo Windows 2000 con el objetivo de que puedan operar con el nuevo protocolo. Si existe algún hosts que no opere con el Windows NT y tenga una plataforma menor, tiene que realizar un upgrade de hardware y memoria para que pueda funcionar con el protocolo IPv6.

6. Para el segundo caso los servidores y el firewall funcionan con el sistema operativo Sun Solaris 2.6, se necesita realizar una actualización de software al Solaris 8.0 el cual opera con el nuevo protocolo IPv6 y cuyos requerimientos mínimos son:

   - 64 MB de memoria en RAM y 110 MB de espacio en disco duro

   No existe inconveniente en el hardware por lo tanto, debe realizar la actualización del software.

Existirán hosts que se mantendrán con el actual protocolo, por lo tanto, ambos protocolos IPv4 e IPv6 deben coexistir. El nuevo protocolo deberá instalarse primero en los routers, después en los servidores, y por último en los hosts, de manera que, a medida que se va cambiando cada componente, éste pueda comunicarse con los que utilicen el protocolo anterior.

Como dato informativo se indican las características principales del nuevo software desarrollado para el manejo de IPv6 de las plataformas anteriormente mencionadas:

- Los equipos ***CISCO*** disponen del software 12.2 el cual soporta IPv6 y tiene las siguientes características:

  - IPv6 unicast routing

- *Servicios*: DNS usando AAAA, TFTP, Tunelamiento automático y manual, path MTU discovery, neighbor discovery, ping, telnet, traceroute
- *Protocolos de capa enlace de datos*: ATM, Ethernet, Fast Ethernet y Gigabit Ethernet, FDDI, Frame Relay, HDLC, LAN virtuales
- *Protocolos de ruteo*: RIP sobre IPv6, rutas estáticas

Cada serie de los routers CISCO debe tener un mínimo de memoria para poder realizar la actualización al nuevo software, se muestra a continuación algunos de ellos:

ROUTER CISCO	SOFTWARE	MEMORIA FLASH (MB)	MEMORIA RAM (MB)
800	12.2.	8	4
805	12.2.	4	8
1400	12.2.	4	16
1601-1604	12.2.	12	4
1720	12.2.	4	24
1750	12.2.	8	32
2501-2525	12.2.	16	10
3660	12.2.	8	32
4500/4700	12.2.	8	32
7500	12.2.	20	128

Fig.4.4 Requerimiento mínimo en memoria flash y RAM para Routers Cisco[166]

Si un router no tiene como mínimo ésta capacidad deberá reemplazar el módulo de memoria por otro nuevo de mayor capacidad.

- *Microsoft Corporation* ha desarrollado un software denominado Microsoft IPv6 Technology Preview para Windows 2000 y que provee aplicaciones como HTTP, FTP y telnet, posee también herramientas como IPv6.exe, ping6.exe, tracert6.exe, ttcp.exe, 6to4cfg.exe, Ipsec6.exe.

[166] http://www.cisco.com/cgi-bin/Software/Iosplanner/Planner-tool/iosplanner.cgi?get_crypto=&data_from=&hardware_name=&software_name=&release_name=&majorRel=12.2&state=

- *Sun Microsystem* ha desarrollado su software Solaris 8 para manejar el nuevo protocolo IPv6 cuyas aplicaciones son:
  - *Sendmail*
  - *Telnet*
  - *Finger*
  - *Tftp*
  - *Snoop*
  - *Route*
  - *Traceroute*
  - *Ping*
  - *BIND*

4.4 COSTOS REFERENCIALES DE LA MIGRACIÓN

En esta sección se realiza un estudio del costo referencial de la migración desde el punto de vista del distribuidor de Internet. Se asume que cada red LAN del cliente corporativo tiene 6 computadores.

Se analizan dos casos:

1. Red con routers marca CISCO, servidores y firewall con sistema operativo Windows NT.

2. Red con routers marca CISCO, servidores y firewall con sistema operativo Sun Solaris.

4.4.1 ANÁLISIS PARA EL PRIMER CASO

El distribuidor requiere hacer el cálculo del costo de la tarifa mensual que debe aplicar al cliente por la actualización al nuevo protocolo, en este caso el distribuidor deberá contratar un préstamo en una entidad bancaria con un costo

del dinero del 18% anual a un plazo de 12 meses (considerando para éste caso que se solicita un préstamo bancario a 1 año).

Existe un componente de costos recurrentes, es decir que se paga mensualmente, en los que se incluye el costo por mantenimiento de routers, servidores y el firewall; todos éstos costos se trasladan al cliente.

El costo final de la tarifa corresponde al costo del pago del préstamo contratado y a los costos recurrentes, más una rentabilidad que para éste caso se considera del 20%, más un margen operativo que se considera del 28%. El detalle de éstos cálculos se indican a continuación:

Parámetros de mensualización: a 12 meses con una tasa de interés del 18%.

	COSTOS GLOBALES	COSTO UNITARIO	CANT	%	TOTAL	MENSUAL	TOTAL/MES
EVENTUALES	ACTUALIZACIÓN DE BACKBONE						
	Licencia por actualización de software por servidor con Windows 2000	$470.00	4		$1,880.00		S/172.36
	Licencia por actualización de routers CISCO R. 12.2	$100.00	2		$200.00		S/18.34
	Actualización de Firewall con S.O. Windows 2000	$470.00	1		$470.00		S/43.09
						SUBTOTAL	S/233.78
EVENTUALES	COSTO POR CLIENTE CORPORATIVO						
	Actualización de software de router clientes corporativos	$100.00	1		$100.00		S/9.17
	Actualización de software por PC (se asumen 6 PCs)	$155.00	6		$930.00		S/85.26
						SUBTOTAL	S/94.43
EVENTUALES	COSTO POR CLIENTE MASIVO						
	Actualización de software por PC	$155.00	1		$155.00		S/14.21

Observación
Se asume que se tienen 4 mantenimientos anuales y que en 1 de ellos se hace el upgrade

	COSTOS RECURRENTES (POR MES)	COSTO UNITARIO	CANT	%	TOTAL	MENSUAL	TOTAL/MES
GLOBALES	Mantenimiento mensual de routers	$80.00	2	25%	$160.00		$40.00
	Mantenimiento mensual de servidores	$100.00	4	25%	$400.00		$100.00
	Mantenimiento mensual de firewall	$90.00	1	25%	$90.00		$22.50
						SUBTOTAL	S/162.50
POR CLIENTE CORPORATIVO	Mantenimiento de router	$80.00	1	25%			$20.00

Fig.4.5 Costos globales y recurrentes de la migración para el primer caso

	GLOBALES	POR CLIENTE CORPORATIVO
RECURRENTES	$162.50	$20.00
EVENTUALES	$233.78	$94.43
SUBTOTAL	$396.28	$114.43
# DE USUARIOS	50	1
COSTO / MES / USUARIO	$7.93	$114.43
TOTAL / MES / USUARIO		$122.36

Fig.4.6 Costos mensualizados por cliente corporativo, primer caso

	GLOBALES	POR CLIENTE MASIVO
RECURRENTES	S/162.50	
EVENTUALES	S/233.78	S/14.21
SUBTOTAL	S/396.28	S/14.21
# DE USUARIOS	500	1
COSTO / MES / USUARIO	S/0.79	S/14.21
TOTAL / MES / USUARIO		S/15.00

Fig.4.7Costos mensualizados por cliente masivo, primer caso

CALCULO DE PRECIO / MES / USUARIO					
	%	CORPORATIVO		MASIVO	
COSTO		S/	122.36	S/	15.00
RENTABILIDAD	20%	S/	24.47	S/	3.00
MARGEN OPERATIVO	28%	S/	34.26	S/	4.20
PRECIO		S/	181.09	S/	22.20

Fig.4.8 Precio a cobrar al cliente, primer caso

De acuerdo con esto, la tarifa a aplicarse al cliente depende del tipo de cliente, es decir al cliente corporativo se le debe cobrar $ 181.09 y al cliente masivo $22.2 mensuales.

4.4.2 ANÁLISIS PARA EL SEGUNDO CASO

El cálculo de la tarifa se lo realiza de la misma manera, la única diferencia radica en los precios de actualización de los servidores y del mantenimiento.

	COSTOS GLOBALES	COSTO UNITARIO	CANT	%	TOTAL	MENSUAL	TOTAL/MES
EVENTUALES	ACTUALIZACIÓN DE BACKBONE						
	Licencia por actualización de software por servidor con Sun Solaris	$100.00	4		$400.00		S/36.67
	Licencia por actualización de routers CISCO	$100.00	2		$200.00		S/18.34
	Actualización de Firewall con S.O. Sun Solaris	$100.00	1		$100.00		S/9.17
						SUBTOTAL	S/64.18

EVENTUALES	COSTO POR CLIENTE CORPORATIVO						
	Actualización de software de router clientes corporativos	$100.00	1		$100.00		S/9.17
	Actualización de software por PC (se asumen 6 PCs)	$155.00	6		$930.00		S/85.26
						SUBTOTAL	S/94.43

EVENTUALES	COSTO POR CLIENTE MASIVO						
	Actualización de software por PC	$155.00	1		$155.00		S/14.21

Observación
Se asume que se tienen 4 mantenimientos anuales y que en 1 de ellos se hace el upgrade

	COSTOS RECURRENTES (POR MES)	COSTO UNITARIO	CANT	%	TOTAL	MENSUAL	TOTAL/MES
GLOBALES	Mantenimiento mensual de routers	$80.00	2	25%	$160.00		$40.00
	Mantenimiento mensual de servidores	$90.00	4	25%	$360.00		$90.00
	Mantenimiento mensual de firewall	$90.00	1	25%	$90.00		$22.50
						SUBTOTAL	S/152.50

POR CLIENTE CORPORATIVO	Mantenimiento de router	$80.00	1	25%			$20.00

Fig.4.9 Costos globales y recurrentes de la migración para el segundo caso

	GLOBALES	POR CLIENTE CORPORATIVO
RECURRENTES	$152.50	$20.00
EVENTUALES	$64.18	$94.43
SUBTOTAL	$216.68	$114.43
# DE USUARIOS	50	1
COSTO / MES / USUARIO	$4.33	$114.43
TOTAL / MES / USUARIO		$118.76

Fig.4.10 Costos mensualizados por cliente corporativo, segundo caso

	GLOBALES	POR CLIENTE MASIVO
RECURRENTES	S/152.50	
EVENTUALES	S/64.18	S/14.21
SUBTOTAL	S/216.68	S/14.21
# DE USUARIOS	500	1
COSTO / MES / USUARIO	S/0.43	S/14.21
TOTAL / MES / USUARIO		S/14.64

Fig.4.11 Costos mensualizados por cliente masivo, segundo caso

CALCULO DE PRECIO / MES / USUARIO					
	%	CORPORATIVO		MASIVO	
COSTO		S/	118.76	S/	14.64
RENTABILIDAD	20%	S/	23.75	S/	2.93
MARGEN OPERATIVO	28%	S/	33.25	S/	4.10
PRECIO		S/	175.77	S/	21.67

Fig.4.11 Precio a cobrar al cliente, segundo caso

La tarifa a aplicarse al cliente corporativo es $ 175.77 y al cliente masivo $21.67 mensuales.

Para los dos casos analizados vemos que el precio a cobrar no tiene una variación grande por la utilización de diferentes tecnologías.

En conclusión el precio que se debe cobrar al cliente dependerá del tipo de equipamiento del que disponga el distribuidor de Internet y del número de clientes; mientras más clientes tenga, el precio a cobrar por la actualización va disminuyendo.

Cabe mencionar que si los equipos utilizan la plataforma de LINUX no tienen costo alguno, ya que este software es gratuito.

Se debe aclarar que este análisis es únicamente para los tipos de configuraciones mencionados anteriormente, en el caso de que se requiera analizar los requerimientos de otro distribuidor de Internet, se debe realizar un estudio específico del tipo de infraestructura que tenga montada.

CAPITULO 5

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES Y RECOMENDACIONES

- A criterio del autor, el objetivo principal de esta tesis se ha cumplido, ya que constituye una guía de consulta sobre conceptos y principios referentes al nuevo Protocolo Internet y los principales problemas de una migración a éste. Se reseña de manera breve el protocolo Internet actual de igual manera el Protocolo Internet versión 6 y los principales métodos de migración.

- La estandarización de IPv6 se comenzó en 1992 pero todavía no se ha llegado a la implementación del protocolo pues aún se encuentra en fase experimental. La existencia de grupos de trabajo destinados al desarrollo del protocolo IPv6 nos hace suponer que la implantación del mismo se llevará a cabo en un futuro cercano.

- IPv4 ha sido usado durante más de 20 años, y en este tiempo se han encontrado conceptos que tienen que ser mejorados. IPv6 incluye una tecnología superior a IPv4 y que es absolutamente imprescindible para el siglo XXI.

- Existen varias razones por las que IPv6 es apropiado para la siguiente generación de Protocolos Internet. Resuelve el problema de adaptación de Internet, provee un mecanismo de transición flexible para la Internet actual, y fue diseñado para encontrar las necesidades de nuevos mercados como dispositivos de computadores personales nomádicos, entretenimiento y control de dispositivos. Hace esto un modo evolutivo que reduce el riesgo de problemas de arquitectura.

- La simplificación de formato de cabecera, algunos campos redundantes de la cabecera IPv4 han sido eliminados o hechos opcionales. Esto reduce el coste de procesamiento de manipulación de paquetes y el ancho de banda de la cabecera. La cabecera IPv6 es solamente dos veces más grande que la de IPv4 considerando que las direcciones son cuatro veces más largas.

- Una dirección IPv6 tiene una longitud de 128 bits, lo que hace que el espacio de direccionamiento sea tan largo que cada persona en el planeta podría tener una red de redes tan extensa como la Internet actual.

- Debido al crecimiento de Internet, el encaminamiento se hace menos manejable con respecto a la eficiencia y requerimientos de memoria, pues aumenta el número de direcciones IP.

- La transición de IPv4 a IPv6 se debe realizar de forma que puedan trabajar al mismo tiempo tanto el protocolo IPv4 como el IPv6. Esto es lógico cuando trabajamos en una red como Internet que abarca un ámbito de operación grande por lo que la transición no se podría realizar en un espacio corto de tiempo.

- El gran problema es que IPv6 es incompatible con la actual versión de IP. Para usar el nuevo protocolo, los administradores de red deberán cambiar el software de protocolos en los dispositivos conectados.

- Como los protocolos forman parte del kernel de los sistemas operativos de muchas máquinas conectadas (como los PCs que corren en UNÍX o en las últimas versiones de Windows NT) cambiar los protocolos IP significaría cambiar también de sistema operativo.

- Una migración a IPv6 podría generar problemas de interoperabilidad en TCP/IP en un futuro.

- Hay ciertos aspectos que no obligan lo suficiente a reemplazar IP por un nuevo protocolo, entre ellos se destacan: nuevo espacio de direcciones para tener más direcciones de red, mas seguridad como criptografía e identificación de usuarios, configuración automática de la red y capacidad para tratar el tráfico sensible a retardos.

- Las tecnologías en las que se basa IPv6 tienen muchos defectos y es muy probable que caigan víctimas de los mismos problemas que ahora afectan a IP.

- Los métodos de migración recomendados por el IETF son la utilización de dos pilas de protocolos y tunneling.

- Con el fin de conseguir un encaminamiento dinámico, surge la necesidad de crear mecanismos que distribuyan globalmente la información de accesibilidad de las capas IPv6 entre regiones dispersas de encaminamiento basado en IPv6.

- El mecanismo básico para el encaminamiento de IPv4 e IPv6 está contenido en el encaminamiento de las dos capas de IP, esto implica que las rutas son calculadas por separado para direcciones IPv4 y para direcciones IPv6.

- La migración a IPv6 requiere que todas las referencias a direcciones de 32 bits sean sustituidas por las nuevas de 128, además, un nuevo tipo de registro deberá ser definido en el DNS para realizar la transformación de los nombres de dominio actuales en direcciones de 128 bits.

- Los firewalls usados con IPv6 analizarán la cabecera para determinar el protocolo de transporte (UDP o TCP) en uso. La función del firewall no debe verse afectada por el uso de IPv6 debido a que el formato de cabecera tiene reglas que facilitan un análisis rápido.

- El backbone de un Distribuidor de Internet tiene varios dispositivos que son de diferentes fabricantes (que es el caso más común). La migración de IPv4 a IPv6 depende exclusivamente del tipo de hardware y software utilizados actualmente en los equipos y si son compatibles con las nuevas versiones desarrolladas por los fabricantes, caso contrario todo el equipamiento deberá ser reemplazado para poder realizar la migración.

- El Backbone de un Distribuidor de Internet debe permitir compatibilidad con cualquier tipo de host sea que funcione con el protocolo IPv4, IPv6 o los dos simultáneamente.

- La mayoría de fabricantes de routers aseguran que la migración de sus equipos solo necesitan una actualización de su software, y que será gratuita para aquellos que tengan soporte directo de fábrica, caso contrario se debe pagar un costo por la actualización.

- Algunos fabricantes de firewalls , servidores y host´s como por ejemplo Microsoft y Sun Solaris, tienen establecido un costo por licencia para la actualización al nuevo software.

- Algunos fabricantes de hosts están desarrollando su software para nuevas versiones como es el caso de la multinacional Microsoft en la cual el software para IPv6 se encuentra sólo para versiones que van del Windows 2000 en adelante, no existe disponibilidad para el Windows 95 ni NT.

- Si una empresa tiene la intención de implementar el protocolo IPv6 en su red, se recomienda primero que los equipos sean probados con este protocolo formando una red paralela para ver su comportamiento, ya que actualmente la mayoría de los equipos están en fase de experimentación.

REFERENCIAS

- Redes Globales de Información con Internet y TCP/IP. Principios básicos, protocolos y arquitectura, Douglas E. Comer, Prentice Hall, Tercera edición 1996.
- Internetworking IPv6 with Cisco Routers, Brown Steven, Mc Graw-Hill, 1997.
- http://www.everex.es/ip.htm
- http://a01-unix.gsyc.inf.uc3m.es/~baudoux/intro.htm
- RFC 1752 The Recommendation for the IP Next Generation Protocol
- RFC 1809 Using the flow Label Field in IPv6
- RFC 1933 Transition Mechanisms for IPv6 Hosts and Routers
- http://www.ietf.org/internet-drafts/draft-iab-case-for-ipv6
- RFC 2460 Internet Protocol, Version 6 (IPv6) Specification. S. Deering, R. Hinden. December 1998.
- RFC 2462 IPv6 Stateless Address Autoconfiguration. S. Thomson, T. Narten. December 1998.
- Internet-draft-ietf-dhcpwg-dhcpv6-00
- Internet-draft-ietf-ipngwg-sec-00
- http://telecom.noc.udg.mx/~kenya/ipv6.html
- http://playground.sun.com/pub/ipng/html/ipng-implementations.html#Nortel
- http://www.cisco.com/ipv6
- http://www.hitachi.co.jp/Prod/comp/network/index.htm
- http://www.nokia.com/ipv6/index.html
- http://playground.sun.com/pub/ipng/html/ipng-implementations.html#TELDAT
- http://playground.sun.com/pub/ipng/html/ipng-implementations.html#3Com
- http://www.microsoft.com/windows2000/technologies/communications/ipv6/default.asp
- http://www.sun.com/solaris/ipv6/;$sessionid$QBA2NUQAAIGQVAMTA1LU4GQ
- http://www.cs-ipv6.lancs.ac.uk/ipv6/systems/linux/faq/

- http://www.software.hp.com/cgi-bin/swdepot_parser.cgi/cgi/displayProductInfo.pl?productNumber=T1306AA
- http://www.compaq.com/info/ipv6/
- http://www.cisco.com/cgi-bin/Software/Iosplanner/Planner-tool/iosplanner.cgi?get_crypto=&data_from=&hardware_name=&software_name=&release_name=&majorRel=12.2&state=

ANEXO A

CODIGOS DE HABILITACIÓN IPv6

ANEXO A

CÓDIGOS DE HABILITACIÓN IPv6 PARA WINDOWS 2000

CÓDIGO DE HABILITACIÓN CLIENTE IPv6

El siguiente código es el agnóstico IP del archivo Client.c, el cual sirve para hablitar la versión IPv6 del archivo Simplec.c.

```
//
// client.c - Simple TCP/UDP client usando Winsock 2.2
//
//      Esta es una muestra del código de fuente de Microsoft.
//      Copyright 1996 - 2000 Microsoft Corporation.
//      Todos los derechos reservados.
//      Este código de fuente está considerado solamente como un suplemento
//      a la documentación de Microsoft Development Tools y/o WinHelp.
//      Vea estas fuentes para información detallada de los programas
//      de ejemplo Microsoft.
//

 define WIN32_LEAN_AND_MEAN
#include <winsock2.h>
#include <ws2tcpip.h>
#ifndef IPPROTO_IPV6
#include <tpipv6.h> // For IPv6 Tech Preview.
#endif
#include <stdlib.h>
#include <stdio.h>
#include <string.h>


//
// Este código asume que se encuentra el nivel de transportel, el sistema solamente
// soporta un stream protocol(TCP) y un datagram protocol (UDP). Además,
// especifica un tipo de socket del SOCK_STREAM que es equivalente a especificar TCP
// y especificar un tipo de socket del SOCK_DGRAM que es equivalente a especificar UDP.
//

#define DEFAULT_SERVER      NULL // Se usará el loopback interface
#define DEFAULT_FAMILY      PF_UNSPEC // Acepta  IPv4 o IPv6
#define DEFAULT_SOCKTYPE    SOCK_STREAM // TCP
#define DEFAULT_PORT        "5001" // Arbitrariamente, realiza un test del puerto
#define DEFAULT_EXTRA       0 // Numero de "extra" bytes a enviar
#define BUFFER_SIZE         65536

void Usage(char *ProgName) {
    fprintf(stderr, "\nSimple socket sample client program.\n");
    fprintf(stderr, "\n%s [-s server] [-f family] [-t transport] [-p port] [-b bytes] [-n
number]\n\n",
            ProgName);
    fprintf(stderr, "  server\tServer name or IP address.  (default: %s)\n",
            (DEFAULT_SERVER == NULL) ? "loopback address" : DEFAULT_SERVER);
```

```
    fprintf(stderr, "  family\tOne of PF_INET, PF_INET6 or PF_UNSPEC.  (default: %s)\n",
            (DEFAULT_FAMILY == PF_UNSPEC) ? "PF_UNSPEC" :
            ((DEFAULT_FAMILY == PF_INET) ? "PF_INET" : "PF_INET6"));
    fprintf(stderr, "  transport\tEither TCP or UDP.  (default: %s)\n",
            (DEFAULT_SOCKTYPE == SOCK_STREAM) ? "TCP" : "UDP");
    fprintf(stderr, "  port\t\tPort on which to connect.  (default: %s)\n",
            DEFAULT_PORT);
    fprintf(stderr, "  bytes\t\tBytes of extra data to send.  (default: %d)\n",
            DEFAULT_EXTRA);
    fprintf(stderr, "  number\tNumber of sends to perform.  (default: 1)\n");
    fprintf(stderr, "  (-n by itself makes client run in an infinite loop,");
    fprintf(stderr, " Hit Ctrl-C to terminate)\n");
    WSACleanup();
    exit(1);
}


LPSTR DecodeError(int ErrorCode)
{
    static char Message[1024];

    // Si este programa fuese multi-threaded, se podría usar
    // FORMAT_MESSAGE_ALLOCATE_BUFFER en lugar de un buffer estático aquí.

    FormatMessage(FORMAT_MESSAGE_FROM_SYSTEM | FORMAT_MESSAGE_IGNORE_INSERTS |
                  FORMAT_MESSAGE_MAX_WIDTH_MASK,
                  NULL, ErrorCode, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
                  (LPSTR)Message, 1024, NULL);
    return Message;
}


int
ReceiveAndPrint(SOCKET ConnSocket, char *Buffer, int BufLen)
{
    int AmountRead;

    AmountRead = recv(ConnSocket, Buffer, BufLen, 0);
    if (AmountRead == SOCKET_ERROR) {
        fprintf(stderr, "recv() failed with error %d: %s\n",
                WSAGetLastError(), DecodeError(WSAGetLastError()));
        closesocket(ConnSocket);
        WSACleanup();
        exit(1);
    }
    //
    // nosostros no vemos esto con UDP, puesto que no hay ninguna
    // conexión establecida
    //
    if (AmountRead == 0) {
        printf("Server closed connection\n");
        closesocket(ConnSocket);
        WSACleanup();
        exit(0);
    }

    printf("Received %d bytes from server: [%.*s]\n",
```

```
            AmountRead, AmountRead, Buffer);

    return AmountRead;
}


int main(int argc, char **argv) {

    char Buffer[BUFFER_SIZE], AddrName[NI_MAXHOST];
    char *Server = DEFAULT_SERVER;
    int Family = DEFAULT_FAMILY;
    int SocketType = DEFAULT_SOCKTYPE;
    char *Port = DEFAULT_PORT;
    int i, RetVal, AddrLen, AmountToSend;
    int ExtraBytes = DEFAULT_EXTRA;
    unsigned int Iteration, MaxIterations = 1;
    BOOL RunForever = FALSE;
    WSADATA wsaData;
    ADDRINFO Hints, *AddrInfo, *AI;
    SOCKET ConnSocket;
    struct sockaddr_storage Addr;

    if (argc > 1) {
        for (i = 1;i < argc; i++) {
            if (((argv[i][0] == '-') || (argv[i][0] == '/')) &&
                (argv[i][1] != 0) && (argv[i][2] == 0)) {
                switch(tolower(argv[i][1])) {
                    case 'f':
                        if (!argv[i+1])
                            Usage(argv[0]);
                        if (!stricmp(argv[i+1], "PF_INET"))
                            Family = PF_INET;
                        else if (!stricmp(argv[i+1], "PF_INET6"))
                            Family = PF_INET6;
                        else if (!stricmp(argv[i+1], "PF_UNSPEC"))
                            Family = PF_UNSPEC;
                        else
                            Usage(argv[0]);
                        i++;
                        break;

                    case 't':
                        if (!argv[i+1])
                            Usage(argv[0]);
                        if (!stricmp(argv[i+1], "TCP"))
                            SocketType = SOCK_STREAM;
                        else if (!stricmp(argv[i+1], "UDP"))
                            SocketType = SOCK_DGRAM;
                        else
                            Usage(argv[0]);
                        i++;
                        break;

                    case 's':
                        if (argv[i+1]) {
                            if (argv[i+1][0] != '-') {
                                Server = argv[++i];
```

```
//
// Por no setear el indicador AI_PASSIVE en las indirectas al getaddrinfo, nosotros
// estamos indicando que intentamos usar el resulatado de la dirección a conectar
// al servicio. Esto significa que cuando el parámetro del servidor es NULO,
// el getaddrinfo volverá una entrada por el protocolo de familia permitido
// conteniendo la dirección de loopback para esta familia.
//

memset(&Hints, 0, sizeof(Hints));
Hints.ai_family = Family;
Hints.ai_socktype = SocketType;
RetVal = getaddrinfo(Server, Port, &Hints, &AddrInfo);
if (RetVal != 0) {
    fprintf(stderr, "Cannot resolve address [%s] and port [%s], error %d: %s\n",
            Server, Port, RetVal, gai_strerror(RetVal));
    WSACleanup();
    return -1;
}

//
// Cada dirección getaddrinfo regresará, hasta que encuentre una a la cual
// se pueda conectar satisfactoriamente.
//
for (AI = AddrInfo; AI != NULL; AI = AI->ai_next) {

    // Open a socket with the correct address family for this address.
    ConnSocket = socket(AI->ai_family, AI->ai_socktype, AI->ai_protocol);
    if (ConnSocket == INVALID_SOCKET) {
        fprintf(stderr,"Error Opening socket, error %d: %s\n",
                WSAGetLastError(), DecodeError(WSAGetLastError()));
        continue;
    }

    //
    // Note que no hay nada en este código que especifique a
    // usar UDP o TCP.
    //
    // Cuando se conecta() está llamando sobre un datagrama socket, esto no está
    // actualmente establecido como una conexión (TCP) socket
    // En lugar, se establece TCP/IP como el medio remoto de los
    // (LocalIPAddress, LocalPort, RemoteIP, RemotePort) mapping.
    // Esto nos habilita a usar send() y recv() sobre datagramas sockets,
    // en lugar de recvfrom() and sendto().
    //

    printf("Attempting to connect to: %s\n", Server ? Server : "localhost");
    if (connect(ConnSocket, AI->ai_addr, AI->ai_addrlen) != SOCKET_ERROR)
        break;

    i = WSAGetLastError();
    if (getnameinfo(AI->ai_addr, AI->ai_addrlen, AddrName,
                    sizeof(AddrName), NULL, 0, NI_NUMERICHOST) != 0)
        strcpy(AddrName, "<unknown>");
    fprintf(stderr, "connect() to %s failed with error %d: %s\n",
            AddrName, i, DecodeError(i));
}
```

```
if (AI == NULL) {
    fprintf(stderr, "Fatal error: unable to connect to the server.\n");
    WSACleanup();
    return -1;
}

//
// Esto demuestra como determinar a donde se conecta un socket.
//
AddrLen = sizeof(Addr);
if (getpeername(ConnSocket, (LPSOCKADDR)&Addr, &AddrLen) == SOCKET_ERROR) {
    fprintf(stderr, "getpeername() failed with error %d: %s\n",
            WSAGetLastError(), DecodeError(WSAGetLastError()));
} else {
    if (getnameinfo((LPSOCKADDR)&Addr, AddrLen, AddrName,
                    sizeof(AddrName), NULL, 0, NI_NUMERICHOST) != 0)
        strcpy(AddrName, "<unknown>");
    printf("Connected to %s, port %d, protocol %s, protocol family %s\n",
           AddrName, ntohs(SS_PORT(&Addr)),
           (AI->ai_socktype == SOCK_STREAM) ? "TCP" : "UDP",
           (AI->ai_family == PF_INET) ? "PF_INET" : "PF_INET6");
}

// Se realiza con la dirección de encadenamiento de información, que puede ser libre.
freeaddrinfo(AddrInfo);

//
// Sin encontrar una dirección local y el puerto del sistema.
//
AddrLen = sizeof(Addr);
if (getsockname(ConnSocket, (LPSOCKADDR)&Addr, &AddrLen) == SOCKET_ERROR) {
    fprintf(stderr, "getsockname() failed with error %d: %s\n",
            WSAGetLastError(), DecodeError(WSAGetLastError()));
} else {
    if (getnameinfo((LPSOCKADDR)&Addr, AddrLen, AddrName,
                    sizeof(AddrName), NULL, 0, NI_NUMERICHOST) != 0)
        strcpy(AddrName, "<unknown>");
    printf("Using local address %s, port %d\n",
           AddrName, ntohs(SS_PORT(&Addr)));
}

//
// Envía y recibe en un loop para el número de requerimientos de iteraciones.
//
for (Iteration = 0; RunForever || Iteration < MaxIterations; Iteration++) {

    // Compose a message to send.
    AmountToSend = sprintf(Buffer, "Message #%u", Iteration + 1);
    for (i = 0; i < ExtraBytes; i++) {
        Buffer[AmountToSend++] = (char)((i & 0x3f) + 0x20);
    }

    // Envía el mensaje.  Puesto que estamos utilizando el bloqueo de un socket, esta
    // llamada no debe volver hasta que envíe entera la cantidad.
    RetVal = send(ConnSocket, Buffer, AmountToSend, 0);
    if (RetVal == SOCKET_ERROR) {
```

```
            fprintf(stderr, "send() failed with error %d: %s\n",
                    WSAGetLastError(), DecodeError(WSAGetLastError()));
            WSACleanup();
            return -1;
        }

        printf("Sent %d bytes (out of %d bytes) of data: [%.*s]\n",
               RetVal, AmountToSend, AmountToSend, Buffer);

        // Borra los buffer justamente para probar que realmente estanos recibiendo algo.
        memset(Buffer, 0, sizeof(Buffer));

        // Recibe e imprime las replicas del servidor.
        ReceiveAndPrint(ConnSocket, Buffer, sizeof(Buffer));
    }

    // Llama al sistema donde estamos enviando.
    printf("Done sending\n");
    shutdown(ConnSocket, SD_SEND);

    //
    // Puesto que el TCP no preserva los límites del mansaje, puede haber
    // más datos que lleguen al servidor. Se continúa recibiendo datos
    // hasta que le servidor cierra la conexión.
    //
    if (SocketType == SOCK_STREAM)
        while(ReceiveAndPrint(ConnSocket, Buffer, sizeof(Buffer)) != 0)
            ;

    closesocket(ConnSocket);
    WSACleanup();
    return 0;
}
```

CÓDIGO DE HABILITACIÓN SERVIDOR IPV6

El siguiente código es el agnóstico IP del archivo Client.c, el cual sirve para hablitar la versión IPv6 del archivo Simplec.c.:

```
//
// server.c - Simple TCP/UDP server using Winsock 2.2
//
//      Esta es una muestra del código de fuente de Microsoft.
//      Copyright 1996 - 2000 Microsoft Corporation.
//      Todos los derechos reservados.
//      Este código de fuente está considerado solamente como un suplemento
//      a la documentación de Microsoft Development Tools y/o WinHelp.
//      Vea estas fuentes para información detallada de los programas
//      de ejemplo Microsoft.
//
```

```
#define WIN32_LEAN_AND_MEAN
#include <winsock2.h>
#include <ws2tcpip.h>
#ifndef IPPROTO_IPV6
#include <tpipv6.h> // For IPv6 Tech Preview.
#endif
#include <stdlib.h>
#include <stdio.h>
#include <string.h>


//
// Este código asume que se encuentra el nivel de transporte, el sistema solamente
// soporta un stream protocol(TCP) y un datagram protocol (UDP). Además,
// especifica un tipo de socket del SOCK_STREAM que es equivalente a especificar TCP
// y especificar un tipo de socket del SOCK_DGRAM que es equivalente a especificar UDP.
//

#define DEFAULT_FAMILY      PF_UNSPEC // AceptA IPv4 o IPv6
#define DEFAULT_SOCKTYPE    SOCK_STREAM // TCP
#define DEFAULT_PORT        "5001" // Arbitrariamente, realiza un test del puerto

#define BUFFER_SIZE         64 // Se setea muy pequeño como demostración

void Usage(char *ProgName) {
    fprintf(stderr, "\nSimple socket sample server program.\n");
    fprintf(stderr, "\n%s [-f family] [-t transport] [-p port] [-a address]\n\n",
            ProgName);
    fprintf(stderr, "  family\tOne of PF_INET, PF_INET6 or PF_UNSPEC.  (default %s)\n",
            (DEFAULT_FAMILY == PF_UNSPEC) ? "PF_UNSPEC" :
            ((DEFAULT_FAMILY == PF_INET) ? "PF_INET" : "PF_INET6"));
    fprintf(stderr, "  transport\tEither TCP or UDP.  (default: %s)\n",
            (DEFAULT_SOCKTYPE == SOCK_STREAM) ? "TCP" : "UDP");
    fprintf(stderr, "  port\t\tPort on which to bind.  (default %s)\n",
            DEFAULT_PORT);
    fprintf(stderr, "  address\tIP address on which to bind.  (default: unspecified address)\n");
    WSACleanup();
    exit(1);
}


LPSTR DecodeError(int ErrorCode)
{
    static char Message[1024];

    // Si este programa fuese multi-threaded, se podría usar
    // FORMAT_MESSAGE_ALLOCATE_BUFFER en lugar de un buffer estático aquí.

    FormatMessage(FORMAT_MESSAGE_FROM_SYSTEM | FORMAT_MESSAGE_IGNORE_INSERTS |
                  FORMAT_MESSAGE_MAX_WIDTH_MASK, NULL, ErrorCode,
                  MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
                  (LPSTR)Message, 1024, NULL);
    return Message;
}


int main(int argc, char **argv)
```

```
{
    char Buffer[BUFFER_SIZE], Hostname[NI_MAXHOST];
    int Family = DEFAULT_FAMILY;
    int SocketType = DEFAULT_SOCKTYPE;
    char *Port = DEFAULT_PORT;
    char *Address = NULL;
    int i, NumSocks, RetVal, FromLen, AmountRead;
    SOCKADDR_STORAGE From;
    WSADATA wsaData;
    ADDRINFO Hints, *AddrInfo, *AI;
    SOCKET ServSock[FD_SETSIZE];
    fd_set SockSet;

    // Parse arguments
    if (argc > 1) {
        for(i = 1;i < argc; i++) {
            if ((argv[i][0] == '-') || (argv[i][0] == '/') &&
                (argv[i][1] != 0) && (argv[i][2] == 0)) {
                switch(tolower(argv[i][1])) {
                    case 'f':
                        if (!argv[i+1])
                            Usage(argv[0]);
                        if (!stricmp(argv[i+1], "PF_INET"))
                            Family = PF_INET;
                        else if (!stricmp(argv[i+1], "PF_INET6"))
                            Family = PF_INET6;
                        else if (!stricmp(argv[i+1], "PF_UNSPEC"))
                            Family = PF_UNSPEC;
                        else
                            Usage(argv[0]);
                        i++;
                        break;

                    case 't':
                        if (!argv[i+1])
                            Usage(argv[0]);
                        if (!stricmp(argv[i+1], "TCP"))
                            SocketType = SOCK_STREAM;
                        else if (!stricmp(argv[i+1], "UDP"))
                            SocketType = SOCK_DGRAM;
                        else
                            Usage(argv[0]);
                        i++;
                        break;

                    case 'a':
                        if (argv[i+1]) {
                            if (argv[i+1][0] != '-') {
                                Address = argv[++i];
                                break;
                            }
                        }
                        Usage(argv[0]);
                        break;

                    case 'p':
                        if (argv[i+1]) {
```

```
                        if (argv[i+1][0] != '-') {
                            Port = argv[++i];
                            break;
                        }
                    }
                    Usage(argv[0]);
                    break;

                default:
                    Usage(argv[0]);
                    break;
            }
        } else
            Usage(argv[0]);
    }
}

// Ask for Winsock version 2.2.
if ((RetVal = WSAStartup(MAKEWORD(2, 2), &wsaData)) != 0) {
    fprintf(stderr, "WSAStartup failed with error %d: %s\n",
            RetVal, DecodeError(RetVal));
    WSACleanup();
    return -1;
}

if (Port == NULL) {
    Usage(argv[0]);
}


//
// Fijando el indicador AI_PASSIVE enlas indirectas al getaddrinfo, estamos
// indicando que nos proponemos utilizar la o las direcciones para atar
// atar a un socket(s) para validar conexiones entrantes.  Esto significa que
// cuando el parámetro de direccionamiento es NULO, el getaddrinfo
// volverá una entrada por la familia permitida del protocolo que contiene la dirección
// sin especificar para esa familia.
//
memset(&Hints, 0, sizeof(Hints));
Hints.ai_family = Family;
Hints.ai_socktype = SocketType;
Hints.ai_flags = AI_NUMERICHOST | AI_PASSIVE;
RetVal = getaddrinfo(Address, Port, &Hints, &AddrInfo);
if (RetVal != 0) {
    fprintf(stderr, "getaddrinfo failed with error %d: %s\n",
            RetVal, gai_strerror(RetVal));
    WSACleanup();
    return -1;
}

//
// Para cada dirección getaddrinfo recobrada, se crea un nuevo socket.
//
for (i = 0, AI = AddrInfo; AI != NULL; AI = AI->ai_next, i++) {

    // Highly unlikely, but check anyway.
    if (i == FD_SETSIZE) {
```

```
            printf("getaddrinfo returned more addresses than we could use.\n");
            break;
        }

        // Este ejemplo solo soporta PF_INET y PF_INET6.
        if ((AI->ai_family != PF_INET) && (AI->ai_family != PF_INET6))
            continue;

        // Abre un socket con la correcta dirección.
        ServSock[i] = socket(AI->ai_family, AI->ai_socktype, AI->ai_protocol);
        if (ServSock[i] == INVALID_SOCKET){
            fprintf(stderr, "socket() failed with error %d: %s\n",
                    WSAGetLastError(), DecodeError(WSAGetLastError()));
            continue;
        }


        //
        // El lazo asocia una combinación del direccionamiento
        // y del acceso con el socket apenas creado. Esto es más útil cuando
        // la aplicación es un servidor que tiene un acceso bien conocido
        // que los clientes conozcan de esto por adelantado.
        //
        if (bind(ServSock[i], AI->ai_addr, AI->ai_addrlen) == SOCKET_ERROR) {
            fprintf(stderr,"bind() failed with error %d: %s\n",
                    WSAGetLastError(), DecodeError(WSAGetLastError()));
            continue;
        }

        //
        // Hasta ahora, todo lo que se ha hecho era aplicable a TCP así como a UDP.
        // Sin embargo, hay diferencias fundamentales entre los protocolos de la secuencia
        // tales como TCP y los protocolos de datagrama tales como UDP.
        //
        // Solamente la conexión orientada a sockets, para estos ejemplos de tipo
        // SOCK_STREAM, pueden escucharse para conexiones entrantes
        //
        if (SocketType == SOCK_STREAM) {
            if (listen(ServSock[i], 5) == SOCKET_ERROR) {
                fprintf(stderr, "listen() failed with error %d: %s\n",
                        WSAGetLastError(), DecodeError(WSAGetLastError()));
                continue;
            }
        }

        printf("'Listening' on port %s, protocol %s, protocol family %s\n",
               Port, (SocketType == SOCK_STREAM) ? "TCP" : "UDP",
               (AI->ai_family == PF_INET) ? "PF_INET" : "PF_INET6");
    }

    freeaddrinfo(AddrInfo);

    if (i == 0) {
        fprintf(stderr, "Fatal error: unable to serve on any address.\n");
        WSACleanup();
        return -1;
    }
```

```
}
if (getnameinfo((LPSOCKADDR)&From, FromLen, Hostname,
                sizeof(Hostname), NULL, 0, NI_NUMERICHOST) != 0)
    strcpy(Hostname, "<unknown>");
printf("\nAccepted connection from %s\n", Hostname);


//
// Este ejemplo de servidor maneja solamente conexiones secuencialmente.
// para manejar conexiones múltiples simultáneamente,un servidor
// desearía probablemente lanzar otra cuerda de rosca o procesarla a este
// punto para manejar cada conexión individual, Alternativamente, podría
// guardar un socket para la conexión y usar()y el uso de
//  fd_set para determinar cual leer del próximo.
//
// Hasta aquí está el loop hasta que la conexión termine.
//

while (1) {

    //
    // Ahora leemos un dato del cliente. Porque TCP
    // NO mantiene límites del mensaje, podemos recv()
    // los datos del cliente agrupados diferentemente que fueron
    // enviados. Puesto que lo que hace este servidor es generación de eco
    // los datos recibidos regresan al cliente, no necesitamos
    // referirnos a los límites del mensaje. Pero significa que los datos
    // del mensaje que imprimimos para un recv() determinado abajo
    // pueden pueden contener más o menos datos que fueron enviados
    // a un determinado cliente.
    //

    AmountRead = recv(ConnSock, Buffer, sizeof(Buffer), 0);
    if (AmountRead == SOCKET_ERROR) {
        fprintf(stderr, "recv() failed with error %d: %s\n",
                WSAGetLastError(), DecodeError(WSAGetLastError()));
        closesocket(ConnSock);
        break;
    }
    if (AmountRead == 0) {
        printf("Client closed connection\n");
        closesocket(ConnSock);
        break;
    }

    printf("Received %d bytes from client: [%.*s]\n",
           AmountRead, AmountRead, Buffer);
    printf("Echoing same data back to client\n");

    RetVal = send(ConnSock, Buffer, AmountRead, 0);
    if (RetVal == SOCKET_ERROR) {
        fprintf(stderr, "send() failed: error %d: %s\n",
                WSAGetLastError(), DecodeError(WSAGetLastError()));
        closesocket(ConnSock);
        break;
    }
}
```

ANEXO B

EJEMPLOS DE DNS IPv6

ANEXO B

EJEMPLOS DE DNS IPv6

Introducción

La información de carácter general sobre BIND (Berkeley Internet Name Domain) se puede encontrar en http://www.isc.org/bind.html . Este sitio también contiene la información adicional sobre la configuración y conexiones del DNS a otros sitios. El resto de este documento asumirá que el programa de lectura es familiar con los aspectos básicos de la configuración del DNS y del BIND. La guía de Glenn Steven es un punto de partida útil si usted no está familiarizado con el DNS. El RFC 1912 es también un documento útil a leer.

BIND (Berkeley Internet Name Domain) es una implementación de los protocolos del Domain Name System (DNS) y proporciona una implementación abierta redistributable de la referencia de los componentes principales del Domain Name System, incluyendo:

- Un servidor Domain Name System (nombrado)
- Una biblioteca del discernidor de imágenes del Domain Name System
- Herramientas para verificar la operación apropiada del servidor del DNS.

El servidor DNS del BIND se utiliza en la mayoría de las máquinas servidoras de nombres sobre la Internet, proporcionando a una configuración robusta y estable encima de la cual la configuración de nombramiento de una organización pueda ser construida. La biblioteca del discernidor de imágenes incluida en la distribución del BIND proporciona al APIs estándar para la traducción entre los

nombres del dominio y los direccionamientos Internet e intenta conectar a las aplicaciones que requieren el servicio conocido.

Cerciórese de que usted esté ejecutando una implementación del BIND (4,9,4 y más allá, preferiblemente las series de versión 8) que es capaz de utilizar los direccionamientos IPv6. Se ponen en ejecución los direccionamientos IPv6 usando un formato de registro del recurso AAAA. Observe que la ayuda de AAAA puede ser indocumentada en algunas versiones antiguas del BIND.

Instalación Básica Del BIND

Para instalar un servidor de DNS, usted debe tener un archivo root cache. Para obtener el archivo actual de la memoria inmediata de la raíz, digite el comando siguiente.

```
cave @ns.internic.net. el ns > / etc/dns/root.cache
```

Si usted obtuvo la distribución completa del BIND, " dig " debe incluirlo dentro de él. Este documento colocará los archivos del DNS en el directorio siguiente.

```
/ etc/dns
```

El item siguiente es disponer correctamente el archivo de configuración del BIND. Las instrucciones específicas de la configuración se pueden encontrar en la guía de las operaciones del BIND (BOG Bind Operations Guide). El BOG debe estar con su distribución BIND. El actual nameserver deamon es un binario llamado "named."

El archivo de configuración del BIND típicamente se llama "named.boot" y está

colocado en el directorio "/ etc" o en el directorio del cual "named " es ejecutado. La primera porción de un archivo de muestra " named.boot ", configurado para la operación IPv6, es como sigue:

```
directory /etc

;type            domain                                    file
cache            .                                         dns/root.cache

; IPV6 NAME LOOKUPS

; Providing backup service for ipv6.vt.edu domain
secondary        ipv6.vt.edu               dns/ipv6/f

; Delegation from ipv6.vt.edu domain
primary          ee.ipv6.vt.edu            dns/ipv6/ee/f
primary          cs.ipv6.vt.edu            dns/ipv6/cs/f
```

Uno puede ver arriba que las entradas delanteras son iguales que IPv4. Las entradas reversas se muestran abajo. Son también similares a IPv4 pero levemente diferente en 1) el direccionamiento invertido es los nibbles del direccionamiento en comparación con los octetos y 2) la clase de dirección es IP6.INT en vez de in-addr.arpa. Observe que usted puede desear dividir el rango de direccionamiento de las entradas reversas diferentes a las de abajo. Es decir, en vez de usar un prefijo / 40 , utilizan un / 64.

```
; IPV6 (REVERSE) ADDRESS LOOKUPS

; VT RFC1897 prefix is 5F05:2000/32
secondary        0.0.0.2.5.0.f.5.IP6.INT          dns/ipv6/r
primary          8.5.0.0.0.0.8.5.d.a.0.8.0.0.0.2.5.0.f.5.IP6.INT
dns/ipv6/ee/r
primary          8.2.0.0.0.0.8.2.d.a.0.8.0.0.0.2.5.0.f.5.IP6.INT
dns/ipv6/cs/r

; Provide address-to-hostname resolution for the loopback address
primary
0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.INT
dns/ipv6/loopback
```

```
forwarders      128.173.4.247 128.173.4.113
```

Si usted setea un secundario (servidor de reserva o alterno), usted substituye " primario " por " secundario " en los ejemplos antedichos.

Observe que las direcciones de arriba están utilizadas usando los direccionamientos originales IPv6 del RFC1897. Estos direccionamientos no son válidos.

Mappings Del Dns

El ejemplo de los mappings delanteros que deberían estar en el archivo "/etc/dns/ipv6/ee/f " es como sigue.

```
; Authoritative data for ee.ipv6.vt.edu
@               IN      SOA ee.ipv6.vt.edu. hostmaster.visc.vt.edu. (
                                97020300        ; Serial (yymmddxx)
                                10800           ; Refresh 3 hours
                                3600            ; Retry   1 hour
                                3600000         ; Expire  1000 hours
                                86400 )         ; Minimum 24 hours
                IN      NS      ocarina.ee.vt.edu.
                IN      NS      moniker.cc.vt.edu.

localhost       IN      A               127.0.0.1

ocarina         IN      A               128.173.88.82
ocarina         IN      AAAA
5F05:2000:80AD:5800:0058:0800:2023:1D71
                IN      HINFO   SPARC5          SOLARIS25
                IN      MX      100     mail.ipv6.vt.edu.

sitar           IN      A               128.173.88.83
sitar           IN      AAAA
        5F05:2000:80AD:5800:0058:0800:2023:2F8E
                IN      HINFO   SPARC5          SOLARIS25
                IN      MX      100     mail.ipv6.vt.edu.


dobro           IN      A               128.173.88.78
dobro           IN      AAAA
5F05:2000:80AD:5800:0058:00AA:00B7:AF2B
                IN      HINFO   P120    WIN95
                IN      MX      100     mail.ipv6.vt.edu.
```

Un ejemplo de los mappings reversos IPv6 para los mappings delanteros antedichos se da abajo. Éste debería estar en el archivo "/ etc/dns/ipv6/ee/r."

```
;       reverse mapping of domain names
;       .8.5.0.0.0.0.8.5.d.a.0.8.0.0.0.2.5.0.f.5.IP6.INT.
;
@               IN      SOA ee.ipv6.vt.edu. hostmaster.vt.edu. (
                                97013000        ; Serial (yymmddxx
format)
                                10800           ; Refresh       3hHours
                                3600            ; Retry         1 hour
                                3600000         ; Expire        1000
hours
                                86400 )         ; Minimum       24 hours
                IN      NS      ocarina.ee.vt.edu.
                IN      NS      moniker.cc.vt.edu.
1.7.d.1.3.2.0.2.0.0.8.0         IN      PTR     ocarina.ee.ipv6.vt.edu.
e.8.f.2.3.2.0.2.0.0.8.0         IN      PTR     sitar.ee.ipv6.vt.edu.
b.2.f.a.3.2.0.2.0.0.8.0         IN      PTR     dobro.ee.ipv6.vt.edu.
```

Verificación

Hay un número de métodos que pueden ser usados para verificar la correcta operación del servidor DNS. Para verificar que su servidor sea un servidor authorststive para el dominio,utilice el siguiente comando.

dig soa +noaa

Un ejemplo es como sigue

```
strat:/home/dlee > dig ece.ipv6.vt.edu soa +noaa

; <<>> DiG 2.2 <<>> ee.ipv6.vt.edu soa +noaa
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6
;; flags: qr aa rd ra; Ques: 1, Ans: 1, Auth: 3, Addit: 3
;; QUESTIONS:
```

```
;;      ee.ipv6.vt.edu, type = SOA, class = IN

;; ANSWERS:
ee.ipv6.vt.edu. 86400   SOA     ee.ipv6.vt.edu. hostmaster.visc.vt.edu. (
                        97091000        ; serial
                        10800   ; refresh (3 hours)
                        3600    ; retry (1 hour)
                        3600000 ; expire (41 days 16 hours)
                        86400 ) ; minimum (1 day)

;; AUTHORITY RECORDS:
ee.ipv6.vt.edu. 86400   NS      dcssvx.cc.vt.edu.
ee.ipv6.vt.edu. 86400   NS      moniker.cc.vt.edu.
ee.ipv6.vt.edu. 86400   NS      zeus.ece.vt.edu.

;; ADDITIONAL RECORDS:
dcssvx.cc.vt.edu.       14400   A       128.173.4.247
moniker.cc.vt.edu.      14400   A       128.173.4.113
zeus.ece.vt.edu.        14400   A       128.173.92.77

;; Total query time: 7 msec
;; FROM: strat to SERVER: default -- 128.173.4.247
;; WHEN: Wed Oct  1 16:43:21 1997
;; MSG SIZE  sent: 32  rcvd: 207
```

Para verificar el direccionamiento individual y las operaciones de búsqueda reversas, se puede utilizar el "nslookup." que es un programa que está también incluido en la distribución BIND. El siguiente ejemplo desarrolla un hostname para el requerimiento de la dirección y entonces una dirección para el requerimiento del hostname "dig" puede también ser usado para verificar la resolución del nombre.

```
strat:/home/dlee  nslookup

Default Server:  dcssvx.cc.vt.edu
Address:  128.173.4.247

 set q=any
 ip6r11.ece.ipv6.vt.edu
Server:  dcssvx.cc.vt.edu
Address:  128.173.4.247

ip6r11.ece.ipv6.vt.edu  IPv6 address = 5f05:2000:80ad:5800::1
ece.ipv6.vt.edu nameserver = dcssvx.cc.vt.edu
ece.ipv6.vt.edu nameserver = moniker.cc.vt.edu
ece.ipv6.vt.edu nameserver = zeus.ece.vt.edu
dcssvx.cc.vt.edu        internet address = 128.173.4.247
moniker.cc.vt.edu       internet address = 128.173.4.113
zeus.ece.vt.edu internet address = 128.173.92.77
```

```
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.5.d.a.0.8.0.0.0.2.5.0.f.5.IP6.INT.
Server:  dcssvx.cc.vt.edu
Address:  128.173.4.247

1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.5.d.a.0.8.0.0.0.2.5.0.f.5.IP6.INT
name = ip6r11.ee.ipv6.vt.edu
0.0.8.5.d.a.0.8.0.0.0.2.5.0.f.5.ip6.int nameserver = dcssvx.cc.vt.edu
0.0.8.5.d.a.0.8.0.0.0.2.5.0.f.5.ip6.int nameserver = moniker.cc.vt.edu
0.0.8.5.d.a.0.8.0.0.0.2.5.0.f.5.ip6.int nameserver = zeus.ece.vt.edu
dcssvx.cc.vt.edu        internet address = 128.173.4.247
moniker.cc.vt.edu       internet address = 128.173.4.113
zeus.ece.vt.edu internet address = 128.173.92.77
```

ANEXO C

CONFIGURACIONES IPv6 CISCO

ANEXO C

CONFIGURACIONES IPv6 CISCO IOS RELEASE 12.2(2)T

Habilitación para Enrutamiento y Configuraciones de Direccionamiento IPv6

Por defecto el enrutamiento IPv6 está deshabilitado en el Software Cisco IOS. Para habilitar el enrutamiento IPv6, primero se debe habilitar el forwarding del tráfico global IPv6 sobre el router y entonces se puede asignar una dirección IPv6 a las interfaces individuales del router.

Habiltando el Proceso Global IPv6 sobre el router

Para habilitar el tráfico global IPv6 sobre el router, se utiliza el siguiente comando en modo de configuración global:

Comando	Propósito
`Router(config) ipv6 unicast-routing`	Habilta los datagramas IPv6 unicast.

Configurando direcciones IPv6

Para configurar una dirección IPv6 sobre una interfaz, use los siguientes comandos, empezando en el modo de configuración global:

	Comando	Propósito
Paso1	`Router(config)#interface` *`interface-type interface-number`*	Especifica el tipo y número de interfaz en modo de configuración
Paso 2	`Router(config-if)#ipv6 address` *`ipv6-prefix/prefix-length`* `[eui-64]`	Especifica una red IPv6 asignada a la interfaz y habilita el proceso IPv6 sobre la interfaz. Especifica la dirección ipv6 *ipv6-prefix/ prefix-length a través del comando de configuración de interfaz sin el* eui-64.
	`Router(config-if)#ipv6 address` *`ipv6-address`* `{`*`/prefix-length`* `| link-local}`	Especifica un direccionamiento IPv6 asignado a la interfazy habilita el proceso IPv6 sobre la interfaz. Especificar la dirección ipv6 *ipv6-address* configurando el comando de interfaz sin el **link-local** keyword configura un site-local y una dirección global IPv6 (La dirección de enlace local para una interfaz es automáticamente configurada cuando IPv6 está habilitada sobre la interfaz.) especificando el comando ipv6 address con el **link-local** keyword se configura un enlace local sobre la interfaz que está usada en ese instante.
	`Router(config-if)#ipv6 unnumbered` *`interface-type interface-number`*	Especifica un interfaz no numerado y habilita el proceso IPv6 sobre la interfaz. El direccionamiento global IPv6 de la interfaz especificado con el argumento *interface-type interface-numberes usado como la dirección fuente de los paquetes generados de la interfaz no numerada..* (Una dirección de enlace local es automáticamente configurada sobre una interfaz no numerada cuando se habilita IPV6 sobre la interfaz.)
	`Router(config-if)#ipv6 enable`	Configura automáticamente una dirección IPv6 de enlace local sobre la interfaz mientras que también habilita el proceso IPv6 sobre la interfaz. La direcciòin de enlace local puede solamente ser usada para comunicarse con nodos del mismo enlace.

Verificando la operación IPv6 y la Configuración de Dirección

Entre el comando ***show running-config*** EXEC para verificar que el proceso IPv6 de paquetes está habilitado globalmente sobre el router y sobre las interfaces aplicables, y que la dirección IPv6 está configurada sobre las interfaces aplicables.

```
Router# show running-config

Building configuration...

Current configuration : 22324 bytes
```

```
!
! Last configuration change at 14:59:38 PST Tue Jan 16 2001
! NVRAM config last updated at 04:25:39 PST Tue Jan 16 2001 by joeblow
!
hostname fred
!

ipv6 unicast-routing
!
interface Ethernet0
 no ip route-cache
 no ip mroute-cache
 no keepalive
 media-type 10BaseT

        ipv6 address 3FFE:C00:0:1::/64 eui-64
!
```

Entre el comando ***show ipv6 interface*** para verificar que las direcciones IPv6 estén configuradas correctamente:

```
Router# show ipv6 interface ethernet 0


Ethernet0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::260:3EFF:FE11:6770
  Global unicast address(es):
    3FFE:C00:0:1:260:3EFF:FE11:6770, subnet is 3FFE:C00:0:1::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF11:6770
  MTU is 1500 bytes
  ICMP error messages limited to one every 500 milliseconds
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
```

Verificando el Acceso a la Lista de Configuración Estándar IPv6

Entre el comando ***show ipv6 acces-list*** EXEC para verificar que la lista de acceso al estándar IPV6 estén configurados correctamente:

```
Router# show ipv6 access-list
```

```
ipv6 access-list florida
 deny 3000::/64 any priority 10
 permit 2000::/64 any priority 20
 permit any any priority 30
```

Entre el comando ***show running-config*** EXEC para verificar que todas las listas de acceso (tanto para IPv4 como para IPV6) estén configuradas correctamente:

```
Router# show running-config

Building configuration...

Current configuration : 22324 bytes
!
! Last configuration change at 14:59:38 PST Tue Jan 16 2001
! NVRAM config last updated at 04:25:39 PST Tue Jan 16 2001 by joeblow
!
hostname fred
!
ipv6 unicast-routing
!
ip classless
!

access-list 2 permit 10.1.1.2
access-list 198 permit ip 172.16.0.0 0.0.255.255 any
access-list 198 permit ip 192.168.0.0 0.0.255.255 any
!

ipv6 access-list florida permit 2000::/64 any priority 10
ipv6 access-list florida deny 3000::/64 any priority 20
ipv6 access-list florida permit any any priority 30
!
```

Configurando Protocolos de Pila Dual y Túneles IPv6

Para soportar la transición de redes solamente IPv4 para integrarse a redes IPv6 e IPv4, el software Cisco IOS soporta los dos protocolos IPv4 e IPv6, y además técnicas de tunneling. Soportando los protocolos de pila dual IPv4 e IPv6 el software de Cisco puede habilitar a una red a recibir y enviar datos con ambos protocolos. Este software soporta configuraciones manuales, automáticas y técnicas de tunelamiento 6 a 4 encapsulando paquetes IPv6 en paquetes IPv4.

Configurando pilas de protocols IPv4 e IPv6

Cuando una interfaz en una red Cisco está configurada con dos direcciones IPv4 e IPv6, la interfaz realiza un forward de los dos tipos de tráfico. La interfaz puede enviar y recibir datos sobre los dos tipos de redes. Para configurar una interfaz en un Cisco éste debe soportar la pila de protocolos IPv4 e IPv6, use el siguiente comando en modo de configuración global:

	Comando	Propósito
Paso 1	`Router(config)#ipv6 unicast-routing`	Habilita datagramas IPv6 unicast
Paso 2	`Router(config)#interface` *`interface-type number`*	Especifica el tipo y número de interfaz
Paso 3	`Router(config-if)#ipv6 address` *`ipv6-prefix /prefix-length`* `[eui-64]`	Especifica la red IPv6 asignada a la interfaz y habilita el proceso IPv6 sobre la.
Paso 4	`Router(config-if)#ip address` *`ip-address mask`* `[secondary]`	Especifica una dirección IPv4 primaria y secundaria sobre la interfaz

Configurando Túneles IPv6

Con los túneles manualmente configurados IPv6, un direccionamiento IPv6 se configura en un interfaz del túnel y los direccionamientos manualmente configurados IPv4 se asignan a la fuente del túnel y al destino del túnel. El host o router en cada extremo de un túnel configurado debe utilizar las pilas del protocolo IPv4 e IPv6.

Con los túneles automáticos IPv6, la fuente del túnel y el destino del túnel son determinadas automáticamente por el direcciones IPv4 en los 32 dígitos binarios de orden inferior de las direcciones IPv4-compatible con IPv6. El host o router en cada extremo de un túnel automático debe utilizar las pilas del protocolo IPv4 e IPv6.

Con los túneles 6to4, el destino del túnel es determinado por la dirección IPv4 del router frontera que se concatena al prefijo 2002::/16 en la dirección de formato 2002:IPv4 del router fronterizo ::/48. El router fronterizo en cada extremo de un túnel 6to4 debe utilizar las pilas del protocolo IPv4 e IPv6.

Configuración manual de un túnel IPv6

	Comando	Propósito
Paso 1	`Router(config)#interface tunnel` *tunnel-number*	Especifica un túnel y un número de interfaz.
Paso 2	`Router(config-if)#ipv6 address` *ipv6-prefix/ prefix-length* `[eui-64]`	Especifica la red IPv6 asignada a la interfaz y habilita el proceso IPv6 sobre la interfaz.
Paso 3	`Router(config-if)#ip address` *ip-address mask* `[secondary]`	Especifica una dirección IPv4 primaria o secundaria para la interfaz.
Paso 4	`Router(config-if)#tunnel source` *{ip-address | interface-type interface-number}*	Especifica la dirección de la fuente IPv4 o el tipo y el número del interfaz de la fuente para el interfaz del túnel .
Paso 5	`Router(config-if)#tunnel destination` *ip-address*	Especifica la dirección del destino IPv4 o el nombre de ordenador principal para el interfaz del túnel .
Paso 6	`Router(config-if)#tunnel mode ipv6ip`	Especifica un túnel manual IPv6.

Configurando un Túnel Automático IPv6

Para configurar un túnel automático IPv6, utilice los comandos siguientes que comienzan en modo de configuración global:

	Comando	Propósito
Paso 1	`Router(config)#interface tunnel` *tunnel-number*	Especifica un interfaz y un número del túnel, y coloca al router en modo de la configuración del interfaz.
Paso 2	`Router(config-if)#tunnel source` *interface-type interface-number*	Especifica el tipo y el número del interfaz de la fuente para el interfaz del túnel. Observe el tipo del interfaz y el número especificado en el comando de la fuente del túnel, se debe configurar con una dirección IPv4 y una dirección IPv6.
Paso3	`Router(config-if)#tunnel mode ipv6ip auto-tunnel`	Especifica un túnel automático IPv6 usando una dirección IPv4-compatible IPv6 .

Configurando un Túnel 6a4

Para configurar un túnel 6to4, utilice los comandos siguientes que comienzan en el modo global de la configuración:

	Comando	Propósito
Paso 1	`Router(config)#interface tunnel` *`tunnel-number`*	Especifica un interfaz y un número del túnel, y coloca al router en modo de la configuración del interfaz.
Paso 2	`Router(config-if)#ipv6 unnumbered` *`interface-type number`*	Permite el proceso de los paquetes IPv6 en el interfaz del túnel sin asignar un direccionamiento explícito IPv6 al interfaz del túnel. Los argumentos del interfaz-tipo y del interfaz-número especifican la dirección de la fuente (direccionamiento global IPv6) que el interfaz innumerable utiliza en los paquetes IPv6 que origina. El direccionamiento de la fuente no puede ser otro interfaz innumerable
Paso 3	`Router(config-if)#tunnel source` *`interface-type interface-number`*	Especifica el tipo del interfaz de la fuente y número para el interfaz del túnel. Observe el tipo del interfaz y el número especificado en el comando de la fuente del túnel debe ser el mismo tipo y número del interfaz especificados en el comando innumerable ipv6. Observe el tipo del interfaz y el número especificado en el comando de la fuente del túnel, se debe configurar con una dirección IPv4 y una dirección IPv6.
Paso 4	`Router(config-if)#tunnel mode ipv6ip 6to4`	Especifica un túnel automático IPv6 usando un direccionamiento 6to4
Paso 5	`Router(config-if)#exit`	Las salidas interconectan el modo de la configuración y retornan all router al modo global de la configuración global
Paso 6	`Router(config)#ipv6 route 2002::/16 tunnel` *`tunnel-number`*	Configura una ruta estática para el prefijo IPv6 2002::/16 al interfaz especificado del túnel.

Verificando la configuración dual del protocol IPv6 y el Tunelamiento

Incorpore el comando ***show running-config*** EXEC para verificar que IPv4 y las pilas del protocolo IPv6 están configurados correctamente en interfaces específicos, y que los túneles del recubrimiento IPv6 están configurados correctamente. En el ejemplo siguiente de la demostración del comando ***show running-config*** , el interfaz 0 de Ethernet y el interfaz 0 del FDDI se configuran con una dirección IPv4 y una dirección IPv6. Además, el interfaz 0 del túnel, el interfaz 1 del túnel, y el interfaz 2 del túnel se configuran como manuales, automáticos, y los túneles 6to4, respectivamente. Una ruta estática IPv6 también se configura para la red 2002::/16 para interconectar el túnel 2 (el túnel 6to4).

```
Router# show running-config

Building configuration...

Current configuration : 22324 bytes
!
hostname fred
!
no ip bootp server
ipv6 unicast-routing
!
interface Tunnel0
 ipv6 address 3ffe:b00:c18:1::3/127
 tunnel source 10.9.14.6
 tunnel destination 172.16.11.21

 tunnel mode ipv6ip
!
interface Tunnel1
 tunnel source Fddi0

 tunnel mode ipv6ip auto-tunnel
!
interface Tunnel2
 ipv6 unnumbered Ethernet0
 tunnel source Ethernet0

tunnel mode ipv6ip 6to4
!
interface Ethernet0

ip address 192.168.99.1 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 no keepalive
 media-type 10BaseT
```

```
 ipv6 enable

ipv6 address 2002:c0a8:6301:1::/64 eui-64
!
interface Fddi0

 ip address 172.31.7.104 255.255.255.224
 no ip route-cache
 no ip mroute-cache
 no keepalive

 ipv6 address 3FFE:C00:0:2::/64 eui-64
!

ipv6 route 2002::/16 Tunnel2
```

Ejemplo de un Túnel IPv6 Configurado Manualmente.

El ejemplo siguiente configura un túnel manual IPv6 entre el router A y el router B. En el ejemplo, el interfaz 0 del túnel para el router A y router B se configura manualmente con un dirección global IPv6 y una dirección IPv4. El destino del túnel también se configura manualmente

Router A Configuration

```
interface tunnel 0

 ipv6 address 3ffe:b00:c18:1::3/127

 ip address 192.168.99.1 255.255.255.0

 tunnel source 192.168.99.1

 tunnel destination 192.168.30.1

 tunnel mode ipv6ip
```

Router B Configuration

```
interface tunnel 0

 ipv6 address 3ffe:b00:c18:1::2/127

 ip address 192.168.30.1 255.255.255.0

 tunnel source 192.168.30.1

 tunnel destination 192.168.99.1

 tunnel mode ipv6ip
```

Ejemplo de Configuración de un Túnel 6a4

El ejemplo siguiente configura un túnel 6to4 entre el router A y el router B. En el ejemplo, el interfaz Ethernet 0 para el router A y router B se configura con una dirección global IPv6 de un ISP a nivel superior y de una dirección IPv4. El interfaz 0 del túnel para el router A y router B se configura sin una dirección IPv4 o IPv6 porque las direcciones IPv4 o IPv6 en el interfaz 0 de Ethernet de ambos routers se utilizan para construir una dirección de la fuente del túnel. Una dirección de destino del túnel no se especifica en cualquier router porque la dirección de destino se construye automáticamente. Una ruta estática IPv6 para la red 2002::/16 para hacer un túnel el interfaz 0 se configura en ambos routers.

Router A Configuration

```
interface Ethernet 0
 ip address 192.168.99.1 255.255.255.0
 ipv6 address 2002:c0a8:6301:1::/64 eui-64
interface Tunnel 0
 ipv6 unnumbered Ethernet 0
 tunnel source Ethernet 0
 tunnel mode ipv6ip 6to4
ipv6 route 2002::/16 Tunnel 0
```

Router B Configuration

```
interface Ethernet 0
 ip address 192.168.30.1 255.255.255.0
 ipv6 address 2002:c0a8:1e01:1::/64 eui-64
interface Tunnel 0
 ipv6 unnumbered Ethernet 0
 tunnel source Ethernet 0
 tunnel mode ipv6ip 6to4
ipv6 route 2002::/16 Tunnel 0
```