ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

ESTUDIO PARA LA MIGRACIÓN DE IPV4 A IPV6 PARA LA EMPRESA PROVEEDORA DE INTERNET MILLTEC S.A.

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

DAVID FERNANDO NUÑEZ LARA
dvngoku@gmail.com

DIRECTOR: Dr. LUIS CORRALES
luisco5049@yahoo.com

Quito, Agosto 2009

DECLARACIÓN

Yo, David Fernando Nuñez Lara, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

David Fernando Nuñez Lara

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por el señor David Fernando Nuñez Lara, bajo mi supervisión.

Dr. Luis Corrales
DIRECTOR DE PROYECTO

AGRADECIMIENTO

A Dios por ser el sustento de mi vida.

A mis padres, hermanas y familiares por su comprensión y apoyo incondicional durante mi vida académica.

A mis amigos por su apoyo en la Universidad.

Al Dr. Luis Corrales por su amistad y ayuda en la dirección de este proyecto.

A los Ingenieros Ramiro Morejón, Carlos Flores, Fernando Flores, Carlos Herrera, y a la Ingeniera Tania Pérez por su ayuda y su amistad.

Y a todos aquellos que de una u otra forma me ayudaron en la culminación de este proyecto.

DEDICATORIA

Al Señor Jesús por darme la vida y la sabiduría para poder culminar esta etapa de mi vida.

A mis padres Gustavo y María, por todo el esfuerzo que hicieron por mi para educarme, apoyarme, enseñarme el valor de las cosas y darme lo mejor en mi vida.

A mis hermanas Andrea, Eliana, Cynthia y Rebeca por dejarme usar la computadora para acabar este proyecto.

A mis tías Rocío y Lupe por toda la ayuda que me brindaron cuando lo necesitaba.

A mis primas Xime y Maty por su apoyo y comprensión porque me ayudaron mucho en la culminación de mis estudios.

A mis amigos Diego (Dieguín), Juan Carlos (Pepa), Geovanny (Lobo), Santiago (San), Leonardo (Lea), Andrés (Pasión), David, Pabel y Christian por ser los mejores amigos y compañeros de trabajo que he tenido, por haber creído en mi y por haberme apoyado y brindado su amistad…Gracias.

Y a todos quienes creyeron en mí.

CONTENIDO

CAPÍTULO 2: ANÁLISIS DE LOS PROTOCOLOS IPV4 E IPV6

CAPÍTULO 3: PLAN PARA LA MIGRACIÓN DE IPV4 A IPV6

CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES

INDICE DE FIGURAS

INDICE DE TABLAS

RESUMEN

Debido a la masificación que ha tenido el Internet, el avance de la tecnología, el desarrollo de las telecomunicaciones y el incremento de la demanda por parte de las empresas se ha producido una escasez de las direcciones IPv4; razón por la cual desde hace muchos años atrás previendo esta situación, se desarrollo el protocolo IPv6 que es un paso evolutivo de IPv4.

Con estos antecedentes, en este proyecto realiza un análisis de la red del proveedor de Internet Milltec S.A. su estructura, los servicios con los que cuenta y los clientes que posee, para luego estudiar las características de IPv6 y diseñar un plan para que el proveedor pueda migrar a IPv6, describiendo también algunos métodos de transición los cuales el ISP podría usar en su proceso de migración a IPv6.

Del estudio realizado se estableció que Milltec S.A. para migrar a IPv6 deberá hacerlo de la siguiente forma: primero la empresa deberá cambiar los routers de los enlaces troncales y el router de la red interna routers que soporten IPv6 en su IOS, luego se deberá realizar las configuraciones necesarias en los servidores y routers para lograr la convivencia entre IPv4 e IPv6 mientras dure el proceso de migración, y por último implementar los servicios y aplicaciones sobre IPv6.

A través de recomendaciones y experiencias que otros ISPs han obtenido al empezar la implementación de IPv6, se concluye que la migración es un proceso que lleva tiempo, pues se compone de muchos pasos como por ejemplo el diseño de un esquema que permita que IPv6 conviva con IPv4; esto quiere decir que los equipos, como los routers, deben estar en la capacidad de trabajar con los dos protocolos. Este proyecto también propone una serie de pasos que se deberían considerar para poner en marcha la migración a IPv6, obtenidos en base a las experiencias y trabajos que muchos de los proveedores de Internet y entidades internacionales dedicadas al desarrollo de IPv6 en el mundo han realizado.

PRESENTACIÓN

En el presente proyecto se realiza un estudio para la migración IPv6 de la red IPv4 de la empresa proveedora de Internet Milltec S.A.

El estudio presenta un análisis de los mecanismos y requerimientos esenciales que se necesitan para que la empresa pueda paulatinamente migrar a IPv6, estudiando primeramente la estructura de la red principal del ISP y luego en base a los conceptos y fundamentos de IPv6 se propone un plan para lograr la migración.

En el Capítulo 1, se presenta un análisis del backbone del ISP, como está estructurado, los servicios que brinda a través de IPv4 y los tipos de clientes con los que actualmente cuenta.

El Capítulo 2, contiene un análisis de IPv6, las características y funcionalidades que actualmente presenta el nuevo protocolo así como una breve comparación con IPv4 identificando las diferencias entre estos dos protocolos. Se estudian y se describen también los diferentes métodos de transición de IPv4 a IPv6 que existen.

El Capítulo 3, presenta un plan para la migración a IPv6 del ISP, basándose en ciertas recomendaciones de entidades internacionales. Se describen configuraciones básicas y necesarias que se requieren en los equipos y servicios que brinda el proveedor, además se presentan de manera tentativa ciertos servicios que el proveedor de Internet podría ofrecer una vez puesto en marcha el plan para la migración.

En el Capítulo 4, se presentan las conclusiones y recomendaciones extraídas del proyecto.

Finalmente, se incluyen los anexos características inherentes a IPv6 como subneting en IPv6, recomendaciones más relevantes de la RFC y algunas configuraciones generales para el DNS y cambios en los sockets de Windows para el soporte de aplicaciones IPv6.

CAPÍTULO 1

ANÁLISIS DE LA SITUACIÓN ACTUAL DE LA EMPRESA PROVEEDORA DE INTERNET MILLTEC S.A.

CAPÍTULO 1

ANÁLISIS DE LA SITUACIÓN ACTUAL DE LA EMPRESA PROVEEDORA DE INTERNET MILLTEC S.A.

1.1 INTRODUCCIÓN

Milltec S.A. es una empresa privada, ubicada en el sector Norte de la ciudad de Quito – Ecuador, la misma que se dedica a proveer servicios de Internet, funcionando solamente dentro del Distrito Metropolitano de Quito.

La empresa, por medio de las aplicaciones en sus equipos comercializa servicios básicos que forman parte de un proveedor de Internet entre sus diferentes tipos de clientes que posee: corporativos, home y dial-up. Los clientes corporativos y home tienen su acceso a la red del ISP a través de conexiones de última milla o bucle local que "es el cableado que se extiende entre la central telefónica y el usuario"[1]. Por otro lado los clientes dial-up usan la Red Telefónica Pública Conmutada (PSTN)[2] para acceder a la red del proveedor.

Las comunicaciones, servicios, y diferentes aplicaciones que el proveedor de Internet brinda son realizados sobre el Protocolo de Internet IP en su versión 4 (IPv4). El trabajo en este proyecto, pretende posteriormente determinar las posibilidades que tiene la empresa frente a una migración del protocolo IPv4 a IPv6.

---

[1] http://es.wikipedia.org/wiki/Bucle_local

[2] Tecnología de conmutación de circuitos que puede utilizarse para implementar una red WAN.

1.2 ESTRUCTURA DE LA RED PRINCIPAL DEL ISP

El backbone está conformado por un router principal que permite el acceso al enlace de salida a Internet[3], cuatro routers con enlaces troncales[4] para el acceso de los clientes a la red del proveedor, servidor de acceso remoto, módems y switches. La Figura 1.1 muestra un esquema de la estructura de la red principal y sus diferentes componentes.

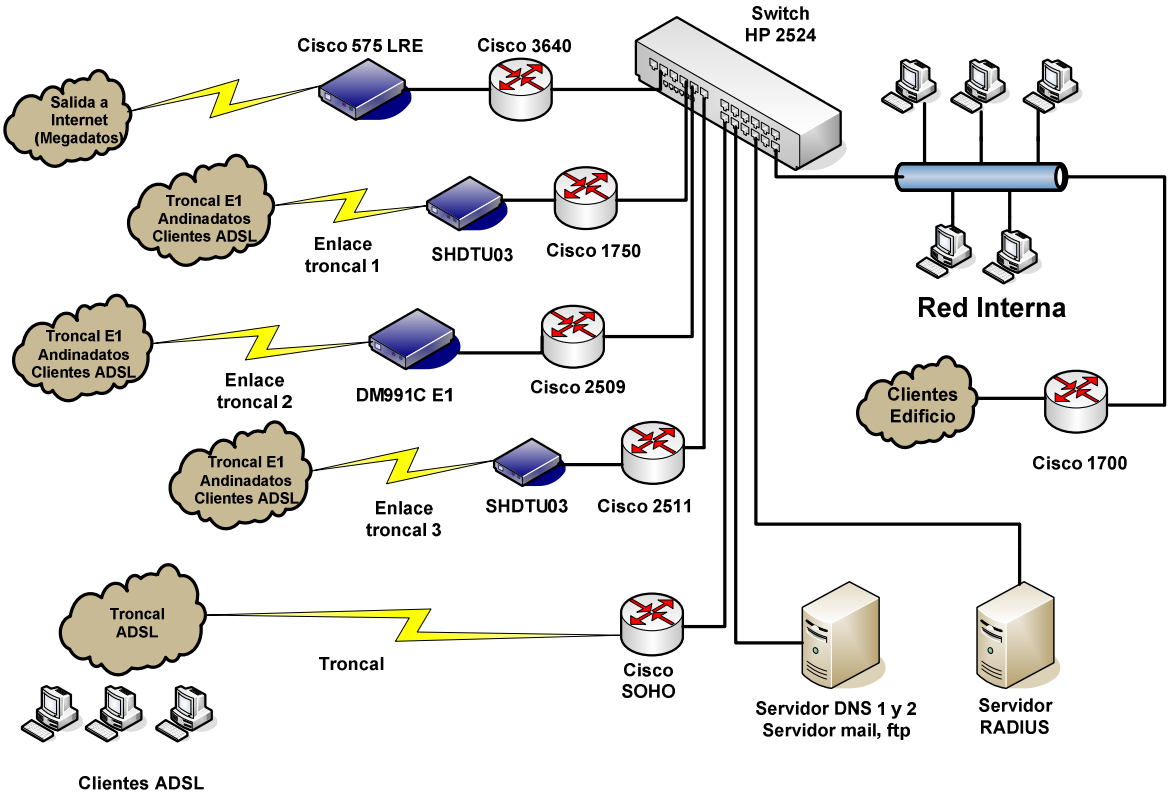

Figura 1.1 Backbone del ISP

[3] El enlace es provisto por la empresa Megadatos.

[4] Cada enlace troncal lo provee La Corporación Nacional de Telecomunicaciones (CNT) y su capacidad equivale a un E1 (2.048 Mbps).

1.2.1 DESCRIPCIÓN DE LA RED DE ACCESO PRINCIPAL

La conforman el router principal y un módem que permite la comunicación entre la salida internacional y el ISP.

1.2.1.1 Router principal

Permite el acceso al Internet mediante un enlace de 5 Mbps provisto por la empresa Megadatos. Es un equipo marca Cisco con las siguientes características:

- **Serie:** Cisco 3600
- **Modelo:** 3640
- **Software:** C3640-TELCO-M
- **Versión:** 12.3(7) T3
- **Procesador:** R4700, 100 MHz
- **Interfaces:** Ethernet, FastEthernet, serial
- **Memoria RAM:** 64 MB
- **Memoria flash:** 16 MB

El router se comunica con Megadatos, a través de un módem (CSU/DSU) que actúa como DCE, por una interfaz FastEthernet que se comunica con la red interna por medio de una interfaz 10/100 BASE-TX.

1.2.1.2 Módem

Este equipo permite la comunicación entre el ISP y el proveedor del enlace de salida al Internet. Es un equipo de marca Cisco 757 LRE, soporta tráfico del servicio telefónico tradicional (POTS) y servicios integrados a través de la red digital (ISDN). El módem puede ser administrado remotamente mediante la interfaz de línea de comandos (CLI)[5] configurando características como: velocidad

[5] Es usado por los equipos Cisco para realizar tareas de configuración mediante comandos.

del enlace, estadísticas y monitoreo de red a través del protocolo SNMP[6]. El proveedor de Internet pone los datos en el enlace a través de un puerto RJ-11, el equipo pertenece a Megadatos.

1.2.2 DESCRIPCIÓN DE LA RED DE ACCESO DE CLIENTES

Compuesta por 4 routers, cada uno soporta un enlace troncal que, mediante módems, permiten el acceso de los clientes a la red del ISP a través de la red ATM[7] de La Corporación Nacional de Telecomunicaciones (CNT).

1.2.2.1 Equipos – enlaces troncales

Permiten el acceso de los clientes a la red del ISP mediante 4 enlaces cuya capacidad es de 2 Mbps cada uno para los tres primeros enlaces, mientras que la capacidad del cuarto enlace es de 0.5 Mbps aproximadamente. Los enlaces son provistos por la Corporación Nacional de Telecomunicaciones. Los equipos ponen los datos en la red a través de sus interfaces seriales.

1.2.2.1.1 Router – enlace troncal 1

El equipo se encarga de enrutar a los clientes al Internet a través de la red del ISP y cumple con las siguientes características:

- **Serie:** Cisco 1700
- **Modelo:** 1750
- **Software:** C1700-SV3Y-M
- **Versión:** 12.1(27a)

---

[6] Protocolo de administración de red simple.

[7] Tecnología que permite la transmisión de servicios como voz, datos y video mediante celdas de longitud fija (53 bytes).

- **Procesador:** M860, 100 MHz
- **Interfaces:** FastEthernet, serial
- **Memoria RAM:** 32 MB
- **Memoria flash:** 8 MB

1.2.2.1.2 Router – enlace troncal 2

Enruta a los clientes al Internet a través de la red del ISP por medio del segundo enlace troncal, el cual trabaja con pocos clientes. El equipo cumple con las siguientes características:

- **Serie:** Cisco 2500
- **Modelo:** 2509
- **Software:** C2500-I-L
- **Versión:** 12.0(7)T
- **Procesador:** Motorola 68EC030, 20 MHZ
- **Interfaces:** Ethernet, serial sincrónica y asincrónica
- **Memoria RAM:** 16 MB
- **Memoria flash:** 4 MB

1.2.2.1.3 Router – enlace troncal 3

El equipo enruta a los clientes al Internet a través de la red del ISP por medio del tercer enlace troncal que trabaja con menos clientes, a diferencia de los enlaces troncales anteriores. Cumple con las siguientes características:

- **Serie:** Cisco 2500
- **Modelo:** 2511
- **Software:** C2500-I-L

- **Versión:** 12.0(3)T
- **Procesador:** Motorola 68EC030, 20 MHz
- **Interfaces:** Ethernet, serial sincrónica y asincrónica
- **Memoria RAM:** 16 MB
- **Memoria flash:** 8 MB

1.2.2.1.4 Cisco SOHO

Este equipo permite el acceso de los clientes ADSL hacia el Internet. Es un equipo Cisco con las siguientes características:

- **Serie:** Cisco SOHO 90
- **Modelo:** 97
- **Software:** SOHO97-K9OY1
- **Versión:** 12.3(8)T
- **Procesador:** Motorola RISC
- **Interfaces:** Ethernet WAN, Ethernet LAN
- **Memoria RAM:** 32 MB
- **Memoria flash:** 8 MB

1.2.2.2 Red Interna

Está compuesta por un router, un switch y los equipos usados por los departamentos que laboran en el ISP, como departamentos de ventas, contabilidad, y el departamento técnico en donde se realiza un monitoreo constante de la red del ISP.

1.2.2.2.1 Router – red local

Dirige el tráfico hacia el Internet tanto del ISP como de los clientes del edificio en donde se encuentran ubicadas las oficinas del proveedor. Es un equipo Cisco y posee las siguientes características:

- **Serie:** Cisco 1700
- **Modelo:** 1700
- **Software:** C1700-SV3Y-M
- **Versión:** 12.1(27a)
- **Procesador:** M860, 100 MHz
- **Interfaces:** FastEthernet, serial
- **Memoria RAM:** 16 MB
- **Memoria flash:** 4 MB

1.2.2.3 Servidores

Son equipos que se dedican a una tarea específica. EL ISP posee dos servidores que cumplen diferentes funciones.

1.2.2.3.1 Servidor 1

Trabaja con el sistema operativo Linux Centos 5; posee un procesador Intel Core 2 DUO, disco duro de 250 GB y memoria RAM de 2GB. Cumple con las funciones de brindar resolución de nombres (DNS) para los equipos del ISP (alberga los DNS primario y secundario del proveedor); brinda servicios de correo soportando los protocolos SMTP y POP3[8], hosting[9], y transferencia de archivos (FTP).

[8] Protocolos que almacenan y envían mensajes de correo; funcionan sobre TCP

[9] Se le denomina hosting al servicio que permite almacenar páginas web en un servidor.

1.2.2.3.2 Servidor 2

Trabaja con un sistema operativo Linux Whitebox, procesador Intel Pentium 4 de 2.7MHz, disco duro de 100GB y memoria RAM de 1GB. Este servidor cumple solamente funciones de un servidor RADIUS, el cual recibe las peticiones de los equipos de acceso, verificando nombres de usuario y contraseñas. Soporta autenticación CHAP y PAP[10].

1.2.3 DESCRIPCIÓN DE LOS ENLACES DEL ISP

El proveedor cuenta con un enlace internacional y tres enlaces troncales los cuales llevan el tráfico de los clientes ADSL y Dial – Up.

1.2.3.1 Enlace internacional

Provisto por la empresa Megadatos, el enlace tiene una capacidad de 5.12Mbps. La administración del tráfico se lo realiza por medio del router Cisco 3640 (se lo describe en la sección 1.2.1.1). El tráfico que es enrutado hacia el Internet debe cursar primero necesariamente por la red de la empresa proveedora del enlace.
El monitoreo del enlace se realiza mediante el MRTG[11] (Multi Router Traffic Grapher).

El monitoreo se lo puede realizar de forma diaria, semanal, mensual o anual. La Figura 1.2 muestra un análisis diario de la cantidad de tráfico; se puede apreciar que el trafico de bajada empieza a alcanzar picos entre 2 Mbps y 4.5 Mbps (dependiendo del día), a partir de las 8:00h, incrementándose de manera paulatina hasta las 18:00h y decrece por las horas de la noche.

---

[10] Protocolos de autenticación que evita el acceso no autorizado.

[11] Herramienta para monitorear el ancho de banda y cuyos datos se obtienen a través del protocolo SNMP.

El tráfico de subida que cursa por el enlace es mucho menor que el de bajada, bordeando los 2 Mbps.

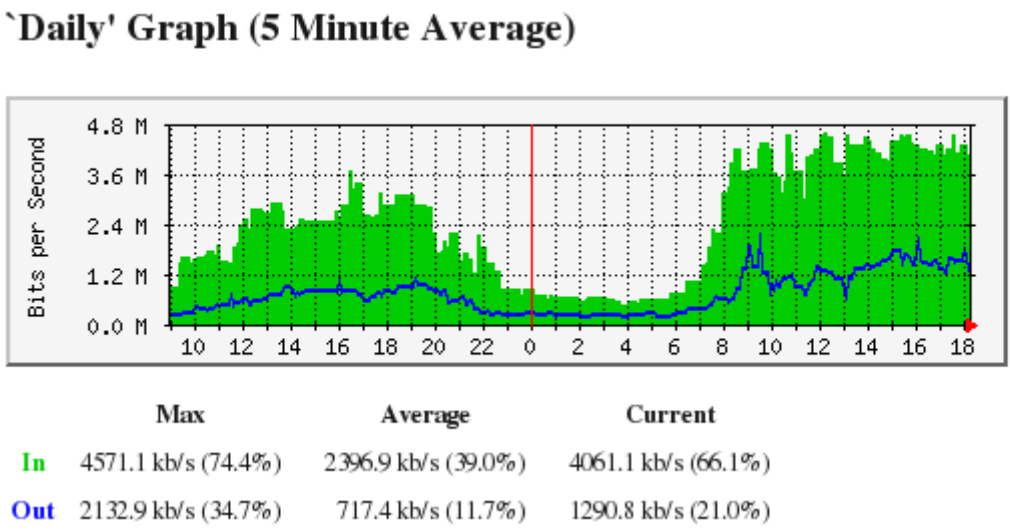

Figura 1.2 Estadística diaria del tráfico que cursa por el enlace internacional

En la Figura 1.3 se puede apreciar estadísticas semanales sobre el tráfico que atraviesa por el enlace; se observa que el tráfico de bajada permanece aproximadamente constante de Lunes a Viernes alcanzando picos entre 4 y 5Mbps, mientras que los fines de semana el tráfico decrece alcanzado aproximadamente los 3 Mbps; en cambio el tráfico de subida alcanza picos entre 2 y 3 Mbps.

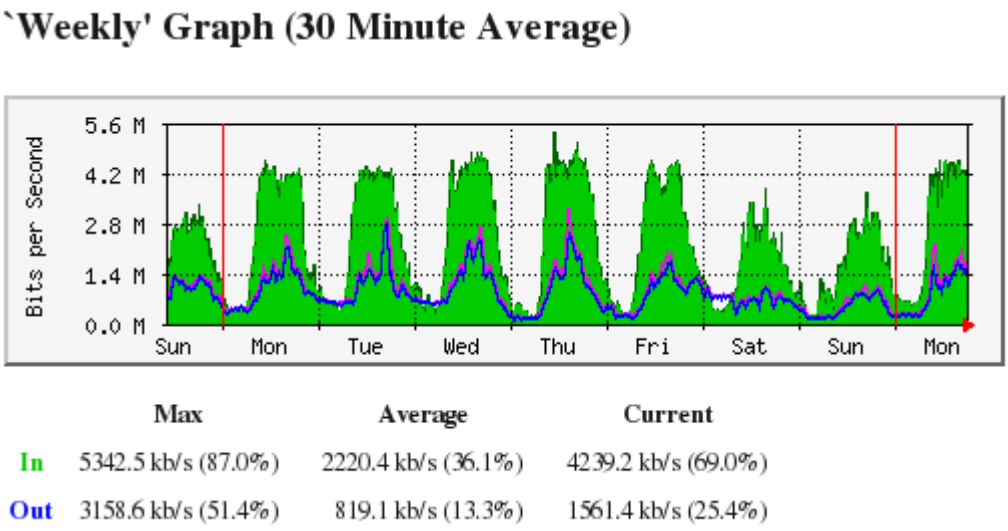

Figura 1.3 Estadística mensual del tráfico que cursa por el enlace internacional

1.2.3.2 Enlaces troncales

Provistos por la CNT, cuentan con una capacidad de 6.5 Mbps, aproximadamente, en total por los cuatro enlaces.

Cada enlace es administrado por un router diferente; estos llevan el tráfico de los clientes corporativos y clientes home del proveedor. El monitoreo de estos enlaces, al igual que en el enlace internacional se lo realiza mediante el MRTG (Multi Router Traffic Grapher).

Se puede apreciar una estadística diaria del tráfico que cursa por el primer E1 en la Figura 1.4. En este enlace se encuentran la mayor parte de los clientes del proveedor; razón por la cual hay una gran cantidad de tráfico alcanzando picos de aproximadamente la capacidad del enlace entre las 8:00h y 18:00h, llegando en ocasiones a saturar el canal.

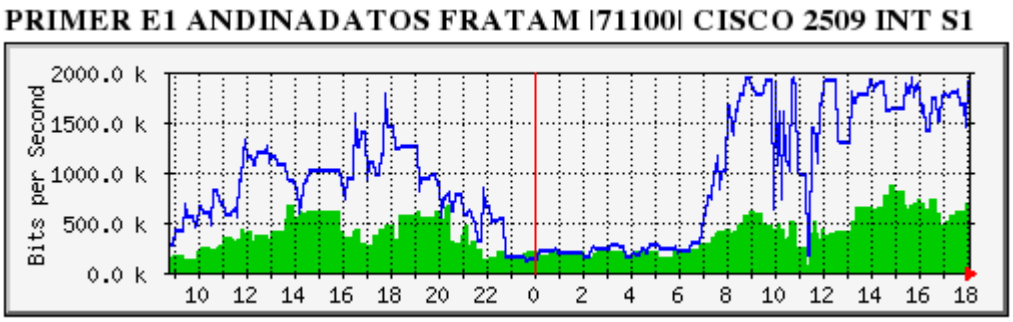

Figura 1.4 Estadística diaria del tráfico que cursa por el primer E1

El tráfico que atraviesa por la segunda troncal es menor en comparación del primer enlace. La Figura 1.5 muestra que los picos son menores a los 2 Mbps, de modo que el canal no permanece saturado como sucede en la primera troncal.

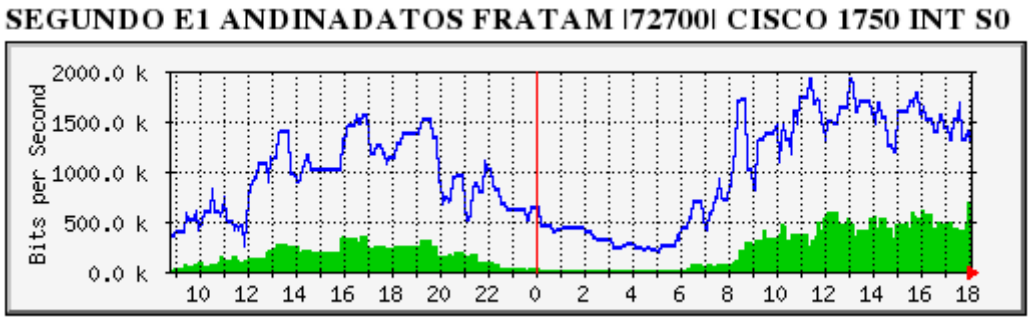

Figura 1.5 Estadística diaria del tráfico que cursa por el segundo E1

La cantidad de clientes que se encuentran en el tercer enlace troncal es mucho menor que los clientes que se encuentran en la primera y segunda troncal, de modo que el canal esta libre y no se satura debido a que solamente alcanza picos de 800 Kbps como se observa en la Figura 1.6; lo que también permite que el proveedor pueda añadir más clientes en este enlace.

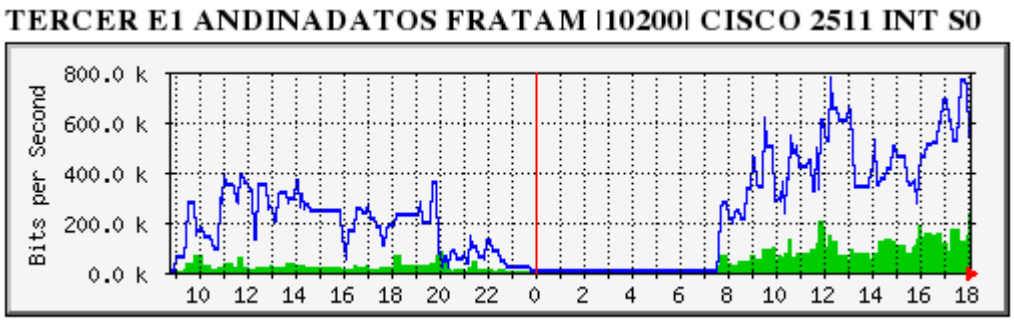

Figura 1.6 Estadística diaria del tráfico que cursa por el tercer E1

La capacidad de la última troncal es de aproximadamente 500 Kbps; sobre esta troncal se encuentran solamente los clientes home. La estadística del tráfico que se observa en la Figura 1.7 indica que se ocupan solo unos 200 Kbps en promedio aproximadamente, ya que dependiendo de la hora y del día pueden haber picos de 312 Kbps o mayores.

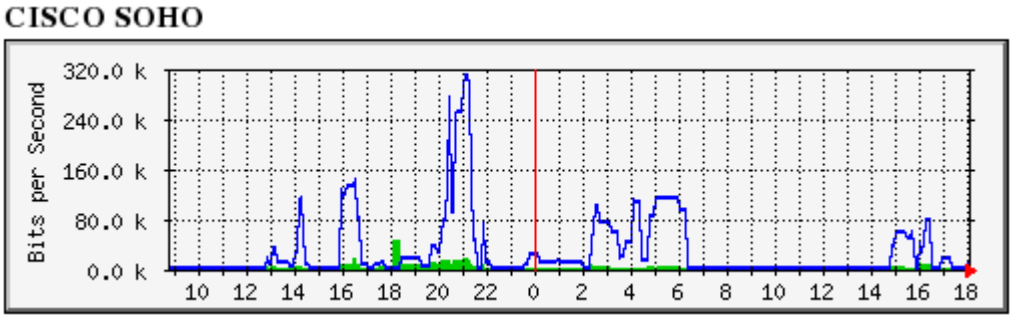

Figura 1.7 Estadística diaria del tráfico que cursa por en enlace troncal

1.3 SERVICIOS

Además del acceso al Internet, el proveedor ofrece servicios de valor agregado a sus clientes, servicios que son comunes en el medio como hosting, transferencia de archivos y servicios de casillas de mail; también ofrece asistencia técnica al cliente en caso de presentarse algún problema con la conexión al Internet.

1.3.1 HOSTING

También es conocido como *“alojamiento web”,* el cual es un servicio que ofrecen comúnmente los proveedores de Internet para almacenar páginas web de clientes o páginas web comerciales; adicionalmente, el proveedor se encarga de comprar un dominio que es un “recurso nemotécnico que se asocian a nodos de la red Internet con el objeto de facilitar su identificación”[12] (dicho de otra forma, es un nombre que se utiliza en lugar de utilizar la dirección IP que es mas difícil de recordar), para que las páginas web puedan ser vistas a través del Internet, mediante un navegador web, usando el protocolo http.

1.3.2 TRANSFERENCIA DE ARCHIVOS

Permite realizar el almacenamiento o descarga de archivos en un servidor a través del protocolo FTP. Dependiendo del tipo de cliente, se asignará una cantidad de espacio de almacenamiento determinada para el uso del cliente.

1.3.3 CORREO ELECTRÓNICO

Permite el envío y la recepción de mensajes de manera rápida; usa el protocolo SMTP para el envío de mensajes y el protocolo POP3 para la recepción.

El proveedor ofrece a todos sus clientes este servicio mediante un domino que es propiedad del ISP. Se asignan dependiendo del tipo de cliente, varias cuentas de correo, cada una de diferente capacidad. Si un cliente requiere una cuenta con un domino diferente, el ISP se encarga de comprar el dominio y proporcionar al cliente sus cuentas de correo con el nuevo dominio.

---

[12] http://es.wikipedia.org/wiki/Dominio_de_Internet

1.4 TIPOS DE CLIENTES

Los clientes del proveedor de Internet son clasificados de acuerdo a la necesidad que tiene el usuario, determinando además los costos correspondientes, según el tipo de cliente y las garantías que el ISP ofrece en el servicio.

1.4.1 CLIENTES CORPORATIVOS

Los clientes tienen acceso a la red del ISP mediante la infraestructura de última milla del proveedor. Pueden contratar velocidades a partir de 64 Kbps simétricas o asimétricas sin compresión alguna; es decir, el canal contratado no lo comparten con ningún otro usuario. Las ventajas que el ISP ofrece a este tipo de clientes son:

- Asignaciones de direcciones públicas
- 25 MB de almacenamiento web
- Número ilimitado de cuentas de correo
- Sopote técnico las 24 horas, incluyendo los fines de semana
- Monitoreo constante
- Antivirus, software gratuito
- Agenda virtual mediante notificaciones por correo electrónico
- Disco duro Virtual, para el almacenamiento de archivos de los clientes

1.4.2 CLIENTES HOME

Tienen acceso a la red del ISP mediante servicios de última milla que son prestados por la CNT. Los clientes pueden contratar velocidades desde 128 Kbps con una compresión de 1:8, compartiendo un canal de 256 Kbps entre varios usuarios, lo que implica que la velocidad contratada es repartida entre el número

de usuarios que compartan el canal. Las ventajas del ISP para estos clientes son las siguientes:

- Cuentas de correo limitadas
- 5MB de espacio para alojamiento web
- Agenda virtual mediante notificaciones por correo electrónico
- Soporte técnico las 24 horas, incluyendo los fines de semana
- Software gratuito
- Monitoreo constante

1.4.3 CLIENTES DIAL – UP

Los clientes usan la Red Telefónica Pública Conmutada (PSNT) para acceder a la red del proveedor quien les otorga a los clientes un nombre de usuario y una contraseña. Estos son autenticados por un servidor de acceso el cual garantiza la conexión al ISP una vez que el cliente haya marcado un número telefónico perteneciente a la PBX del ISP. Las velocidades de acceso que tienen las conexiones Dial – up son de 56 Kbps o más bajas. Las ventajas con las que cuentan los clientes son las siguientes:

- Un casillero de e-mail
- Soporte técnico las 24 horas, incluyendo los fines de semana
- Monitoreo constante

Cabe mencionar que el ISP solamente cuenta con clientes corporativos y home; en la actualidad los clientes Dial-up han sido cedidos a Milltec S.A. pero la migración total de los clientes sigue en proceso ya que no ha sido aún completada.

En el presente capítulo se ha descrito la situación actual del ISP, la estructura y funcionamiento de su red principal. También se describieron los servicios que el

mismo brinda a través de IPv4. En el capítulo siguiente se verán los fundamentos, características fundamentales y funcionalidades de IPv4 e IPv6, además de los mecanismos de transmisión de IPv4 a IPv6, lo cual constituye un elemento principal e importante para la migración.

CAPÍTULO 2

ANÁLISIS DE LOS PROTOCOLOS IPV4 E IPV6

CAPÍTULO 2

ANÁLISIS DE LOS PROTOCOLOS IPV4 E IPV6

2.1 INTRODUCCIÓN

El protocolo IP fue desarrollado en 1973 junto con el protocolo TCP, como parte de un proyecto patrocinado por la Agencia de Programas Avanzados de Investigación (ARPA) del departamento de Defensa de los Estados Unidos (DoD). Se encuentra en la capa de red del modelo OSI, es un protocolo no orientado a conexión y no confiable; cada paquete es tratado de manera independiente de todos los demás y la entrega de paquetes no se garantiza.

El protocolo IP permite la interconexión de redes, proporcionando un esquema de transporte para el envío de paquetes desde un origen a un destino sin importar si se encuentran o no en diferentes redes. El envío de paquetes lo realiza a través de la transmisión de bloques de datos conocidos como datagramas. El origen y destino son identificados mediante direcciones fijas conocidas como direcciones IP en donde cada dispositivo debe tener una dirección única.

Existen dos versiones del protocolo IP a nivel de capa de red que actualmente están siendo usadas; la versión 4 (IPv4) y la versión 6 (IPv6). Esta última fue desarrollada debido a la gran masificación que ha tenido el Internet en el mundo global provocando el agotamiento de direcciones IPv4.

IPv6 está siendo implementada en algunas áreas debido a las exigencias que son cada vez mayores por el fuerte crecimiento y desarrollo del Internet, además de la

evolución de las redes actuales. IPv6 llegará a reemplazar paulatinamente a IPv4.

2.2 LIMITACIONES DE IPV4

La versión 4 del protocolo IP no ha cambiado desde la recomendación RFC 791 que fue publicada en 1981, sin embargo, la evolución del Internet ha sido sustancialmente grande y a pesar del crecimiento exponencial que este ha tenido, el diseño inicial de IPv4 no anticipo ciertos detalles como:

- ***El agotamiento de las direcciones IP***. Aunque los 32 bits que posee una dirección IPv4 permiten tener 4.294'967.296 millones de direcciones, la expansión del Internet ha hecho que estas direcciones se vuelvan cada vez más escasas. Esto ha llevado a algunas organizaciones a usar un traductor de direcciones (NAT) para asignar una dirección pública a varias direcciones privadas promoviendo la reutilización de las direcciones privadas. Si bien esta medida ha ayudado a evitar el agotamiento de las direcciones IP, NAT genera cuellos de botellas en la comunicación.

- ***La necesidad de una simple configuración***. Debido al crecimiento del Internet también ha aumentado el uso de computadores y dispositivos que requieren una dirección IP. Esto ha dado origen a la necesidad de una configuración automática y más simple de direcciones, además de otras opciones de configuración, que no se basen en una infraestructura manual ni de DHCP.

- ***Los requerimientos de seguridad a nivel de IP***. A pesar de que en la actualidad existe la norma IPSEC (Protocolo de Internet de seguridad) para garantizar la seguridad para los paquetes IPv4; este estándar es opcional para IPv4 y las comunicaciones privadas sobre un medio

público, como lo es el Internet, requieren de servicios de seguridad que les permita mantener la confidencialidad y la integridad sus datos.

- ***La necesidad de un mejor soporte en la entrega en tiempo real de los datos.*** Aunque existen estándares para QoS en IPv4, el soporte del tráfico en tiempo real se basa en los 8 bits del campo TOS (Tipo de servicio) de IPv4 usando por lo general el protocolo TCP o UDP; pero, desafortunadamente, el campo TOS ha sido limitado y con el tiempo ha sido redefinido para uso local. Además, la identificación de la carga útil que usa un puerto TCP o UDP no es posible cuando la carga útil del paquete IPv4 está encriptada.

2.3 DESCRIPCIÓN DE IPV6

IPv6 fue inicialmente desarrollado a principios de los años 90 debido a la necesidad creciente de más direcciones IP. Debido a los grandes recursos que empezaba a ofrecer el Internet, se empezaron a diseñar aparatos como teléfonos móviles, electrodomésticos inteligentes, nuevas tecnologías y aplicaciones que podían brindar nuevos servicios a través del protocolo IP.

Para retardar el agotamiento de las direcciones, se creó NAT (Network Address Translation) el cual permite que varios usuarios se comuniquen al Internet a través de una sola dirección IP. Pero como ya se mencionó, esta solución genera cuellos de botella, lo cual pone limitaciones a la comunicación. Otro problema que presenta el uso de NAT es que en cierto tipo de comunicaciones no se puede encriptar la información si así lo requería una institución, debido a que NAT reemplaza las direcciones y los puertos para realizar la traducción. En consecuencia, existe un impacto en la aplicación de protocolos de seguridad para el envío de información cuando se usa NAT.

La IETF está trabajando para hacer que IPv6 sea más robusto y más eficiente a través de mejoras en áreas como enrutamiento y auto-configuración de red. Los

nuevos dispositivos que pueden conectarse al Internet, y que ya están en el mercado son dispositivos “plug-and-play”, es decir, que se autoconfiguran[13] las direcciones como si se tuviera un sistema DHCP pero para IPv6. De esta manera los dispositivos generalmente a través de una funcionalidad llamada “Routing Discovery”, buscarán el camino a Internet por si solos, haciendo que no sea necesario configurar parámetros como el gateway o máscara de subred.

Varias corporaciones, empresas y agencias de gobiernos de distintos países han sido capaces de lograr varias mejoras en IPv6 de modo que el protocolo en la actualidad es capaz de proveer confiabilidad, escalabilidad, calidad de servicio (QoS) y una entrega sólida de los datos y la información de extremo a extremo para servicios como voz sobre IP (VoIP), IPTV y para redes triple play.

2.3.1 CARACTERÍSTICAS DE IPV6

IPv6 posee características esenciales como:

- ***Un nuevo formato de cabecera.*** Fue diseñado para reducir el trabajo que realizan los equipos de enrutamiento al momento de procesar la información.

- ***Espacio más grande para las direcciones.*** IPv6 posee un espacio de 128 bits para las direcciones de origen y de destino lo que permite tener 3.4×10^{38} posibles direcciones, a diferencia de IPv4 que solamente ocupa 32 bits.

- ***Direccionamiento eficiente y jerárquico.*** IPv6 posee una estructura de enrutamiento eficiente y jerárquica que permite a los routers principales que trabajan en el Internet poseer tablas de enrutamiento más pequeñas, de acuerdo a la infraestructura que tenga cada ISP.

[13] La referencia IPV200501 hace alusión a varios términos para IPv6; ver Anexo 1.

- ***Autoconfiguración.*** Las direcciones IPv6 pueden ser configuradas manualmente o automáticamente, aún en la ausencia de un router. Esto debido a que los hosts pueden autoconfigurarse con enlaces de direcciones locales (local-link addresses) sin la necesidad de configuración manual.

- ***Seguridad.*** IPv6 posee características de seguridad como encripción, de la carga útil (payload) y la autenticación de la fuente de la comunicación.

- ***QoS.*** El tráfico es priorizado mediante un campo de clase de tráfico (Class Traffic). Un campo en la cabecera IPv6 permite a los routers identificar y proporcionar un tratamiento especial a los paquetes que pertenecen a un determinado flujo[14].

- ***Interacción con nodos vecinos.*** Mediante un protocolo llamado Neighbor Discovery, que posee IPv6, se puede manejar una serie de mensajes IPv6 que permitan la interacción de nodos vecinos.

- ***Extensibilidad:*** IPv6 puede ser fácilmente modificado, ya que añadiendo extensiones a la cabecera IPv6 se pueden obtener nuevas características para el protocolo.

2.3.2 IPV4 FRENTE A IPV6

IPv6 mantiene varias funciones usadas en IPv4; en cambio, funciones que eran usadas en muy pocas ocasiones o no eran usadas han sido eliminadas. Esto permite añadirle a este nuevo protocolo nuevas características que provean nuevas funcionalidades para la comunicación a través del Internet.

---

[14] Serie de paquetes que son intercambiados entre un origen y un destino.

Las diferencias más importantes entre IPv4 e IPv6 se muestran en la Tabla 2.1 a continuación:

IPv4	IPv6
Las direcciones de origen y destino tienen una longitud de 32 bits (4 bytes).	Las direcciones de origen y destino tienen una longitud de 128 bits (16 bytes).
La implementación de IPSec es opcional.	La implementación y soporte para IPSec es obligatorio.
Ninguna identificación de flujo de paquete para QoS es manejada por los routers en la cabecera de IPv4.	La Identificación de flujo de paquete para QoS está presente en la cabecera IPv6 usando el campo "flow Label"
La fragmentación es realizada en IPv4 involucra tanto al host como el router, de modo que este proceso produce retardos en el rendimiento del router.	El proceso de fragmentación en IPv6 solamente involucra al host ya que el paquete es procesado solo en el nodo final de destino.
No tiene ningún requisito para el tamaño de un paquete de capa de enlace y debe ser capaz de reensamblar un paquete de 576 bytes.	La capa de enlace de soportar un paquete de 1280 bytes de tamaño y debe ser capaz de reensamblar un paquete de 1500 bytes.
La cabecera incluye el checksum.	La cabecera no incluye Checksum.
La cabecera incluye campos llamados opciones.	Todos los datos opcionales son movidos a las cabeceras extendidas que tiene IPv6.
ARP envía tramas broadcast para realizar peticiones ARP de modo que se pueda resolver una dirección IPv4 en una dirección de capa física.	Las tramas para solicitar peticiones ARP son reemplazadas con mensajes multicast "Neighbor Discovery".
IGMP (Internet Group Management Protocol) es usado para manejar grupos de subredes locales.	IGMP es reemplazado por MLD (Multicast Listener Discovery) que es un set de mensajes que son intercambiados por los routers para descubrir direcciones multicast.
ICMP Router Discovery es usado para determinar la dirección IPv4 del mejor "gateway" y es opcional.	ICMPv4 es reemplazado por mensajes ICMPv6 y es necesariamente requerido.
Las direcciones de broadcast son utilizadas para enviar tráfico a todos los nodos en una subred.	No existen direcciones IPv6 de broadcast, en su lugar los enlaces locales echan una mirada en todos los nodos en donde direcciones multicast son usadas.
Las direcciones deben ser configuradas manualmente o mediante DHCP.	Las direcciones IPv6 no requieren configuración manual o DHCP.
Usa recursos de registros de direcciones de host in DNS para asignar nombres a direcciones IP.	Usa registros AAAA in DNS para asignar nombres a direcciones IPv6.

Tabla 2.1 Diferencias entre IPv4 e IPv6.

2.4 DIRECCIONAMIENTO IPV6

Una dirección IPv6 tiene una longitud de 128 bits divididos en bloques de 16 bits donde cada bloque es representado por 4 dígitos hexadecimales, a diferencia de IPv4 en donde los grupos de 8 bits eran representados por dígitos decimales. Cada bloque de 4 dígitos hexadecimales, es separado por el signo ":" mientras que en Ipv4 la separación de los bloques se la realiza con el signo ".".

Existen reglas que pueden ser aplicadas a las direcciones IPv6 con el objetivo de resumir un poco la sintaxis de las direcciones. Por ejemplo una dirección IPv6 válida 2001:0000:1234:0000:0000:C1C0:ABCD:0876 puede aceptar lo siguiente:

- Las letras pueden ser mayúsculas o minúsculas y la dirección se puede escribir como 2001:0000:1234:0000:0000:**c**1**c**0:**abcd**:0876

- Los "ceros" consecutivos son opcionales y se los puede representar en la dirección como 2001:**0**:1234:**0**:**0**:C1C0:ABCD:**0**876

- Los campos sucesivos de "ceros" pueden ser reemplazados por "::" y la dirección puede tomar la forma 2001:**0**:1234**::**C1C0:**abcd**:876. Pero, cualquier dirección que tenga mas de una vez la representación "::" será una dirección inválida ya que solamente se puede usar esa representación una sola vez.

2.4.1 PREFIJOS

El prefijo se emplea en las direcciones con el formato <dirección> / <longitud del prefijo> donde la longitud del prefijo en IPv4 equivalía a la longitud de la máscara de la subred para separarla de la porción de host.

En una dirección IPv6 un host que por ejemplo tiene la dirección 3ffe:b00:c18:1::1/64 identifica a los primeros 64 bits (debido a /64) como el

número de red y los 64 bits restantes como parte del host. Cualquier prefijo que sea menor a 64 bits es una ruta o un rango de direcciones que son resumidos de una porción del espacio de direcciones IPv6.

2.4.2 TIPOS DE DIRECCIONES IPV6

Existen 3 tipos de direcciones IPv6: unicast, multicast y anycast.

2.4.2.1 Direcciones Unicast[15]

Identifica una interfaz única en el ámbito de direcciones. Los paquetes que son dirigidos a una dirección unicast son entregados en una interfaz única.

Para dar cabida a los sistemas de balanceo de carga la norma RFC 2373 permite a múltiples interfaces utilizar la misma dirección, siempre y cuando aparezcan como una solo interfaz para la implementación de IPv6 en el host. Las direcciones unicast tienen diferentes tipos de direcciones como direcciones globales unicast, link-local, site-local, direcciones especiales, direcciones compatibles.

2.4.2.1.1 Direcciones globales

Las direcciones unicast globales en IPv6 son equivalentes a las direcciones públicas en IPv4. Estas direcciones son enrutables y accesibles a nivel global sobre la porción de IPv6 en Internet.

Estas direcciones están diseñadas para ser agregadas o sumarizadas para producir una estructura eficiente de enrutamiento.

---

[15] DAVIES Jhosep. Understanding IPv6, Washington USA, página 60.

2.4.2.1.2 Direcciones link-local

Son usadas por los nodos que se comunican con los nodos vecinos que se encuentran en el mismo enlace; por ejemplo, en un único enlace IPv6 sin router, las direcciones link-local (locales de enlace) se utilizan para la comunicación entre los hosts dentro del enlace.

Las direcciones link-local siempre comienzan con FE80; con la interfaz de 64 bits de identificación. El prefijo para las direcciones link-local es siempre FE80::/64.

2.4.2.1.3 Direcciones site-local

Las direcciones site-local son equivalentes a las direcciones privadas IPv4 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16. Las intranets privadas que no tienen una conexión directa al Internet a través de IPv6 pueden usar direcciones site-local sin entrar en conflicto con las direcciones globales. Las direcciones site-local no son accesibles desde otros sitios y los routers no deben enviar tráfico site-local fuera del enlace local.

2.4.2.1.4 Direcciones IPv6 especiales

Existen dos direcciones especiales que son usadas en IPv6 y son las direcciones no especificadas y las direcciones de loopback.

Las direcciones no especificadas (0:0:0:0:0:0:0:0 ó ::) son usadas para identificar la ausencia de una dirección. Esta dirección es típicamente usada como una dirección de origen cuando una dirección no ha sido aún determinada. Nunca se la asigna a una interfaz o es usada como dirección de destino.

Las direcciones de loopback (0:0:0:0:0:0:0:1 ó ::1) es usada para identificar una interfaz de loopback, habilitando a un nodo para que pueda enviarse paquetes a sí mismo.

2.4.2.1.5 Direcciones compatibles

Sirven para ayudar en la migración de IPv4 a IPv6, para la coexistencia de ambas direcciones y entre las cuales estan: direcciones IPv4 compatibles, direcciones 6over4, direcciones 6to4 y direcciones ISATAP.

Las direcciones IPv4 compatibles son usadas por nodos IPv6/IPv4 que se comunican con IPv6 sobre una infraestructura IPv6 pública.

Las direcciones 6over4 son usadas para representar a un host en el mecanismo de entunelamiento (tunneling) conocido como 6over4 el cual; junto con otros mecanismos de migración, se lo estudiará posteriormente.

Las direcciones 6to4 se las utiliza para representar un nodo en el mecanismo de entunelamiento 6to4.

Las direcciones ISATAP son usadas para representar a un nodo para el mecanismo de asignación de direcciones conocido como Intra-Site Automatic Tunnel Addresing Protocol (ISATAP).

2.4.2.2 Direcciones Multicast[16]

Las direcciones multicast habilitan el uso eficiente del ancho de banda de la red al enviar un mínimo número de datagramas al número máximo de nodos. En un enlace local un datagrama multicast es enviado a un ilimitado número de nodos;

[16] DAVIES Jhosep. Understanding IPv6, Washington USA, página 70.

un prefijo especial (en IPv4 es 224.0.0.0/8) identifica a un datagrama multicast y una dirección específica dentro del prefijo identifica a cada grupo de nodos. La Tabla 2.2 indica las direcciones IPv6 que son reservadas para multicast.

Direcciones IPv6 Multicast	Descripción
FF02::1	Todas las direcciones de todos los nodos son usadas para alcanzar a todos los nodos en el mismo enlace.
FF02::2	Todas las direcciones de los routers son usadas para alcanzar a todos los routers en el mismo enlace.
FF02::4	La dirección es usada para alcanzar a todos los protocolos de enrutamiento multicast de vector distancia (DVMRP) que usan los routers multicast en el mismo enlace.
FF02::5	La dirección es usada para alcanzar a todos los routers que usan OPSF en el mismo enlace.
FF02::1:FFXX:XXXX	La dirección solicitada del nodo es usada en el proceso de resolución de direcciones para resolver la dirección IPv6 de un nodo en el mismo enlace, a una dirección de capa de red. Los 24 bits últimos de la derecha de la dirección del nodo, son los mismos 24 bits últimos de la derecha de una dirección unicast.

Tabla 2.2 Direcciones reservadas para Multicast.

Las direcciones de multicast no pueden ser usadas como direcciones de origen o como destinos intermediarios. Las direcciones multicast incluyen una estructura adicional para identificar las banderas, su ámbito de aplicación y a que grupo multicast pertenece.

2.4.2.3 Direcciones Anycast[17]

Una dirección Anycast es asignada a múltiples interfaces. Los destinatarios de los paquetes de una dirección anycast se transmiten por la infraestructura de enrutamiento más cercana a la interfaz a la que la dirección anycast es asignada.

Con el fin de facilitar la entrega, la infraestructura de enrutamiento debe saber la distancia de las interfaces que tienen direcciones anycast en términos de métrica de enrutamiento. Este conocimiento se logra a través de la propagación de las rutas de acogida en toda la infraestructura de enrutamiento de la porción de red que no puede resumir la dirección anycast mediante una ruta prefijo.

2.4.3 DIRECCIONES IPV6 PARA HOST Y ROUTER

Por lo general, a un elemento de red que cuenta con una sola interfaz de red se le puede asignar tan solamente una dirección IPv4, pero a uno que usa IPv6 se le puede asignar múltiples direcciones por cada interfaz y las direcciones que se le puede asignar a un host o router que usa IPv6 son las siguientes direcciones unicast:

- Una dirección link-local por cada interfaz
- Una dirección unicast adicional (que puede ser site-local o dirección global) por cada interfaz.
- La dirección de loopback (::1) para la interfaz de loopback.

Los hosts y routers IPv6 tienen al menos 2 direcciones; además cada interfaz está pendiente de escuchar el tráfico multicast.

---

[17] DAVIES Jhosep. Understanding IPv6, Washington USA, página 72

2.4.4 IDENTIFICADORES DE INTERFACE IPV6

En IPv4 la ID de un host o nodo está dada por una porción de la dirección IPv4 la misma que es un identificador lógico de una interfaz en una subred IPv4. Por otro lado en IPv6 la ID está compuesta por los últimos 64 bits de la dirección IPv6; esto es para acomodar la dirección con los 48 bits de una dirección MAC, la misma que es usada en la mayoría de tecnologías LAN, como Ethernet por ejemplo.

Las formas en las que se determina un identificador de interfaz son las siguientes:

- Tal y como se define en la recomendación RFC 2373, todas la direcciones unicast que utilizan el prefijo 001 por el 111 también deben utilizar una interfaz de 64 bits de identificación (EUI-64). Los 64 bits de la dirección EUI-64 están definidos por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE); las direcciones son asignadas a un adaptador de red o se derivan de las direcciones IEEE 802.

- Tal y como se define en la recomendación RFC 3041 se puede tener un identificador de interfaz temporalmente asignado y aleatoriamente generado para proporcionar un nivel de anonimato.

- Tal y como se define en la recomendación RFC 2472 un identificador de interfaz puede estar basado en la capa de enlace, números de serie o números que se generan aleatoriamente cuando se configura una interfaz con el protocolo PPP y la dirección EUI-64 no esta disponible.

- El identificador de interfaz es asignado durante la configuración manual de la dirección.

2.4.5 DIRECCIONES IPV4 E IPV6 EQUIVALENTES

Para resumir la relación entre el direccionamiento IPv4 y el direccionamiento IPv6 la Tabla 2.3 muestra los conceptos de direccionamiento IPv4 y sus equivalentes en IPv6.

Direcciones IPv4	***Direcciones IPv6***
Internet address classes	No es aplicable en IPv6
Direcciones Multicast (224.0.0.0/4)	Dirección multicast IPv6 (FF00::/8)
Dirección de Broadcast	No es aplicable en IPv6
La dirección no especificada es 0.0.0.0	La dirección no especificada es ::
La dirección de Loopback es 127.0.0.1	La dirección de Loopback es ::1
Direcciones IP públicas	Direcciones globales unicast
Dirección IP privada (10.0.0.0/8,)	Direcciones Site-local (FEC0::/48) 172.16.0.0/12, and 192.168.0.0/16)
La representación de las direcciones se realiza con notación decimal.	La representación de las direcciónes se la realiza con notación hexadecimal y la supresión de los ceros.
La máscara de subred se representa con notación decimal con puntos o longitud del prefijo.	Los bits de red se representan solo con la longitud del prefijo.

Tabla 2.3 Direcciones IPv4 equivalentes.

2.5 ICMPv6

Al igual que IPv4, las especificaciones para las cabeceras IPv6 y las cabeceras extendidas IPv6 no ofrecen un reporte confiable de errores. Para que esto sea más factible IPv6 utiliza una versión actualizada del protocolo ICMP llamado también ICMPv6, el cual tiene funciones similares de ICMPv4 tales como presentación de informes de entrega y transmisión de errores.

ICMPv6 provee una estructura para las siguientes aplicaciones de este protocolo: *Neighbor Discovery* y *Multicast Listener Discovery*.

2.5.1 NEIGHBOR DISCOVERY[18]

Es una serie de 5 mensajes ICMPv6 que gestionan la comunicación nodo a nodo en un enlace y determinan las relaciones entre los nodos vecinos. Neighbor Discovery reemplaza a ARP y a los mensajes "*Router Discovery*" y "*Redirect message*" de ICMPv4.

Neighbor Discovery es usado por los nodos para:

- Resolver la dirección de capa de enlace de un vecino al cual un paquete IPv6 esta siendo enviado.
- Determinar cuando la dirección de capa de enlace de un vecino ha cambiado.
- Determinar si un vecino sigue aún siendo alcanzable.

Neighbor Discovery es usado por los hosts para:

- Descubrir los routers vecinos.
- Autoconfigurar direcciones, prefijos de direcciones, rutas y otros parámetros de configuración.

Neighbor Discovery es usado por los routers para:

- Advertir su presencia, parámetros de configuración de los hosts y rutas.
- Informar a los hosts sobre una mejor ruta para el envío de paquetes a un destino específico.

[18] DAVIES Jhosep. Understanding IPv6, Washington USA, página 85

2.5.1.1 Procesos[19]

Los procesos que Neighbor Discovery incluye son los siguientes:

- **Router Discovery:** Durante este proceso, un host descubre los routers que se encuentran en un enlace local. El proceso es similar al que efectúa Router Discovery en ICMPv4.
- **Prefix Discovery:** Mediante este proceso los hosts descubren los prefijos de red para los destino en un enlace local.
- **Parámetros:** Este proceso permite que los host puedan descubrir parámetros adicionales que se encuentran en operación como por ejemplo el límite de saltos que pueden realizar los paquetes salientes.
- **Autoconfiguración de direcciones:** Durante el proceso de autoconfiguración las direcciones IP son configuradas en ausencia o presencia de un servidor DHCP.
- **Resolución de direcciones:** En este proceso, los nodos resuelven direcciones IPv6 a direcciones de Capa 2 de un vecino.
- **Determinación del próximo salto:** Durante este proceso un nodo determina la dirección IPv6 del vecino al cual un paquete está siendo enviado, basándose en la dirección de destino.
- **Detección de dirección duplicada:** Durante el proceso de detección de una dirección duplicada, un nodo determina si una dirección considerada para su uso está siendo usada por un nodo vecino.
- **Función de re-dirección:** Este proceso informa a un host de un mejor "primer salto" a través de una dirección IPv6 para alcanzar su destino.

2.5.1.2 Tipos de mensajes[20]

Todas las funciones que Neighbor Discovery presenta en IPv6 están desarrolladas en los siguientes tipos de mensajes: solicitud de router, Aviso de

---

[19] DAVIES Jhosep. Understanding IPv6, Washington USA, página 88

[20] DAVIES Jhosep. Understanding IPv6, Washington USA, página 90.

router, solicitud de vecino, aviso de vecino y redirección las mismas que son descritas brevemente a continuación:

- **Solicitud de router:** Este mensaje es enviado por todos los hosts IPv6 para descubrir la presencia de router IPv6 en el enlace. Un host envía un mensaje multicast para la solicitud el mismo que más tarde será contestado por los routers IPv6 a través del mensaje “aviso de router”.
- **Aviso de router:** este mensaje es enviado en respuesta al mensaje “solicitud de router”, y contiene toda la información solicitada por el host como por ejemplo prefijos de direcciones, MTU, rutas específicas, si se está o no usando autoconfiguración en las direcciones, etc.
- **Solicitud de vecino:** Es un mensaje enviado por los hosts IPv6 para descubrir la dirección de capa 2 de un enlace activo en un nodo IPv6 donde generalmente se incluye la dirección de Capa 2 del solicitante. Estos mensajes son multicast para la resolución de direcciones y son unicast cuando se está verificando si un nodo vecino puede ser alcanzable.
- **Aviso de vecino:** Este mensaje es enviado en respuesta al mensaje “solicitud de vecino”. Este mensaje contiene información requerida por los nodos para determinar el tipo de aviso del mensaje, el rol en la red del host que esta realizando una solicitud y generalmente la dirección de Capa 2 del host de origen.
- **Redirección:** Este mensaje es enviado por un router IPv6 para informar al host de origen sobre un mejor salto que puede tener un paquete para un destino específico. Este tipo de mensajes son solamente enviados por los routers como tráfico unicast y son solo procesados por los hosts.

2.5.2 MULTICAST LISTENER

Multicast Listener Discovery (MLD) que es usado en IPv6 es el equivalente de Internet Group Management Protocol (IGMPv2) usado en IPv4. MLD es un conjunto de mensajes que son intercambiados por hosts y routers, que permite a

los routers descubrir un conjunto de direcciones multicast las mismas que son escuchas por los hosts en la correspondiente subred a la cual pertenecen.

Los mensajes MLD son enviados con una dirección IPv6 de origen; el límite saltos es siempre 1, y de esta forma se evita que los mensajes sean enviados por un router.

2.5.2.1 Procesos[21]

Existen dos procesos involucrados en los mensajes MLD, el primero es cuando el host se une a un grupo multicast y el segundo cuando deja el grupo.

Primero un host se une a un grupo multicast cuando envía un mensaje MLD destinado a una dirección multicast de interés. Un router es configurado para que acepte todos los paquetes multicast provenientes de un enlace; el router reconoce los paquetes MLD mediante la opción “Router Alert” (compone un campo del formato de mensaje MLD) y pasa este mensaje a la capa superior. Un host que es un miembro de un grupo también recibe un mensaje de reporte enviado al grupo multicast por otro host que está uniéndose al grupo. Luego el host que está recibiendo el mensaje recuerda que hay alguien que quiere unirse al grupo. La existencia de otros miembros del grupo afecta al siguiente proceso de MLD.

Cuando un host va a dejar un grupo multicast y es el último en enviar un mensaje MLD, notifica al router sobre la salida enviando un mensaje “Multicast Listener Done” a todos los router multicast del grupo. En el caso de que el host saliente detecte la presencia de otro miembro en el grupo que está escuchando las direcciones multicast, el mensaje Done no es generado e indicada al router que existe otro miembro presente; en el caso en que el mensaje “Done” es generado, el router envía al host que está saliendo una serie de mensajes de espera, y si efectivamente el host al cual se envía los mensajes de espera está saliendo del

[21] SHIMA Keiichi. IPv6 Advanced Protocols Implementación, San Francisco USA, página 114.

grupo, el router no recibe ninguna respuesta y cesa de enviar paquetes multicast una vez que las transmisiones de los mensajes de espera se completen sin que el router haya recibido respuesta alguna.

2.5.3 MENSAJES ICMPv6

Existen 2 tipos de mensajes para ICMPv6: mensajes de error y mensajes de información.

2.5.3.1 Mensajes de error

Estos reportan los errores de envío y de entrega por un router cualquiera o por un host de destino. Para conservar el ancho de banda de la red estos mensaje no son enviados cada vez que es detectado un error, en lugar de eso los mensajes ICMP v6 son limitados por un temporizador que esta fijado para que un mensaje de error sea enviado cada 7 milisegundos y por un porcentaje correspondiente al 2% del ancho de banda.

Los siguientes mensajes ICMPv6 de error están constituidos por los siguientes mensajes:

- **Destino Inalcanzable (Mensaje ICMPv6 tipo 1):** Un router o un host de destino envían este tipo de mensaje cuando el paquete no puede ser enviado al destino o a un protocolo de capa superior.
- **Paquete demasiado grande (Mensaje ICMPv6 tipo 2):** Este mensaje es enviado cuando el paquete no puede ser transmitido porque el MTU en la interface de salida de un router es más pequeña que el tamaño del paquete IPv6.
- **Tiempo Excedido:** Generalmente es un router el que envía este tipo de mensaje cuando el límite de saltos en la cabecera IPv6 llega a ser cero después de decrementar su valor durante el proceso de envío.

- **Problema de parámetro:** Este mensaje es enviado o por un router o por el destino y ocurre cuando hay un error en la cabecera IPv6 o en una cabecera extendida.

2.5.3.2 Mensajes de información

Estos mensajes proveen la capacidad de dar un simple diagnóstico y ayudar en la búsqueda de problemas; además algunos de los mensajes que componen los mensajes de información de ICMPv6 son usados por Neighbor Discovery y Multicast Listener.

Los mensajes de información de ICMPv6 están compuestos por “Echo Request” y “Echo Reply”.

- **Echo Request:** Este mensaje es enviado a un destino solicitando inmediatamente un mensaje “Echo Reply”. Tanto “Echo Request” como “Echo Reply” proveen un diagnóstico simple para la ayuda en la búsqueda de problemas de enrutamiento.
- **Echo Reply:** Este mensaje es enviado en respuesta a “Echo Reply”.

2.6 IPV6 ROUTING

La creación de una red IPv6 se compone de múltiples subredes IPv6 conectadas entre ellas por los routers IPv6. Para proporcionar accesibilidad a cualquier ubicación dentro de la red IPv6 las rutas a través de hosts y routers deben existir para enviar el tráfico deseado al destino. Estas rutas pueden ser rutas generales, tales como las rutas por defecto, las cuales sumarizan todas la localizaciones. También pueden ser rutas específicas, así como las rutas de subred, las que sumarizan todas las localizaciones de una subred específica.

Los hosts suelen utilizar rutas directamente conectadas para llegar a los nodos vecinos y una ruta por defecto para llegar a los otros destinos. Los routers suelen usar rutas específicas para llegar a todos los lugares dentro de su ubicación, y utilizan rutas sumarizadas para llegar a otros sitios o al Internet. Aunque la configuración de la ruta por defecto y de las rutas para comunicar a los hosts con las redes directamente conectadas o las redes remotas son hechas de manera automática con el mensaje “Router Advertisement”. La configuración de los routers en si es más compleja ya que un router puede tener rutas configuradas dinámicamente o mediante la configuración de protocolos de enrutamiento.

De la misma forma como se emplea el enrutamiento en IPv4, en IPv6 es necesario tener una tabla de enrutamiento para poder determinar como hacer el reenvío de paquetes.

2.6.1 TABLA DE ERUTAMIENTO IPv6[22]

La tabla de enrutamiento almacena información sobre redes IPv6 y como se puede llegar a ellas. Cada dispositivo que implementa IPv6 determina la forma en como se transmiten los paquetes basándose en el contenido de la tabla de enrutamiento en IPv6. La tabla de enrutamiento contiene la siguiente información:

- Prefijo de la dirección.
- La interface sobre la cual las PDU que coinciden con el prefijo de la dirección, son enviadas.
- La dirección de próximo salto.
- Un valor usado para seleccionar entre múltiples rutas con los mismos prefijos.
- El tiempo de vida de la ruta.
- La información acerca de la publicación de la ruta.

[22] AMOSS John, MINOLI Daniel. IPv4 to IPv6 Transition, New York USA, página 40.

- La expiración de la ruta.
- El tipo de ruta.

La tabla de enrutamiento IPv6 se construye automáticamente sobre la actual configuración IPv6 que se tenga en los routers. Cuando se reenvían PDUs IPv6 el router busca en la tabla de enrutamiento alguna entrada con la coincidencia más especifica para la dirección IPv6 de destino.

Una ruta por defecto es usada por un dispositivo final porque no es práctico para el dispositivo mantener una tabla de enrutamiento para cada comunicación dentro de una red IPv6. El prefijo de una ruta por defecto es ::/0, y es generalmente usada para enviar un PDU IPv6 al router principal del enlace local ya que ese router posee información acerca de los prefijos de la red de otras subredes IPv6. La PDU es enviada a otros routers hasta que finalmente es entregada al destino.

2.6.1.1 Proceso de enrutamiento[23]

Los siguientes procesos ocurren durante el enrutamiento:

- Antes de que el dispositivo que está iniciando la comunicación envíe un paquete IPv6, inserta su dirección IPv6 de origen para el destinatario, en la cabecera IPv6.
- A continuación el dispositivo de destino examina la dirección IPv6 de destino y la compara con su tabla local de enrutamiento y luego realiza una de las siguientes acciones:
  - Pasa la PDU a un protocolo IPv6 de capa superior en el host local.
  - Envía la PDU a través de una de las interfaces de red directamente conectadas.
  - Descarta la PDU.

[23] AMOSS John, MINOLI Daniel. IPv4 to IPv6 Transition, New York USA, página 50.

- IPv6 busca en la tabla de enrutamiento la ruta que está más cercana a la dirección IPv6 de destino. La ruta más específica o menos específica es determinada en el siguiente orden:
  - Una ruta que coincida con la dirección IPv6 de destino (una ruta de host con un prefijo de 128 bits de longitud).
  - Una ruta en la que coincida el destino con el prefijo de longitud más largo.
  - La ruta por defecto (su prefijo es ::/0)

- Si la ruta que coincida no es encontrada entonces el destino es determinado como un destino “on-link”.

2.6.2 TIPOS DE ENRUTAMIENTO

2.6.2.1 Estático

El enrutamiento estático es basado en las entradas de la tabla de enrutamiento que son manualmente configuradas y no cambian con un cambio en la topología de red. Un router cuyas tablas de enrutamiento han sido configuradas manualmente se lo conoce como router estático. Los routers estáticos pueden funcionar muy bien en redes pequeñas pero no en redes de gran escala, o en redes que tienen cambios dinámicos, ya que su administración se la realiza manualmente.

Un administrador de red que conoce acerca de la topología de la red puede manualmente construir y actualizar la tabla de enrutamiento ingresando todas las rutas en la tabla.

2.6.2.2 Dinámico

En el enrutamiento dinámico las entradas de la tabla de enrutamiento se actualizan de forma automática para los cambios dados en la topología de red. Un router cuyas tablas de enrutamiento se han configurado para que funcionen dinámicamente se lo conoce como router dinámico. Las tablas de estos routers son construidas y mantenidas automáticamente a través de la comunicación permanente entre los routers. Esta comunicación se ve facilitada por un protocolo de enrutamiento que emplea una serie de mensajes periódicos con información que solo es intercambiada entre los routers. Los routers dinámicos requieren poco mantenimiento y pueden funcionar en redes de gran escala.

2.6.3 PROTOCOLOS DE ENRUTAMIENTO

Un protocolo de enrutamiento es usado para facilitar el intercambio de información concerniente a rutas entre routers.

Los protocolos de enrutamiento poseen un elemento importante para detectar y recuperarse de fallas en la red; la información de enrutamiento que se propaga a través de la red. Cuando todos los routers tienen la información correcta de enrutamiento en sus tablas, entonces se puede decir que la red ha convergido y una vez que se logra la convergencia la red se encuentra en un estado estable.

La clasificación de los protocolos de enrutamiento dinámicos está representada en la Figura 2.1

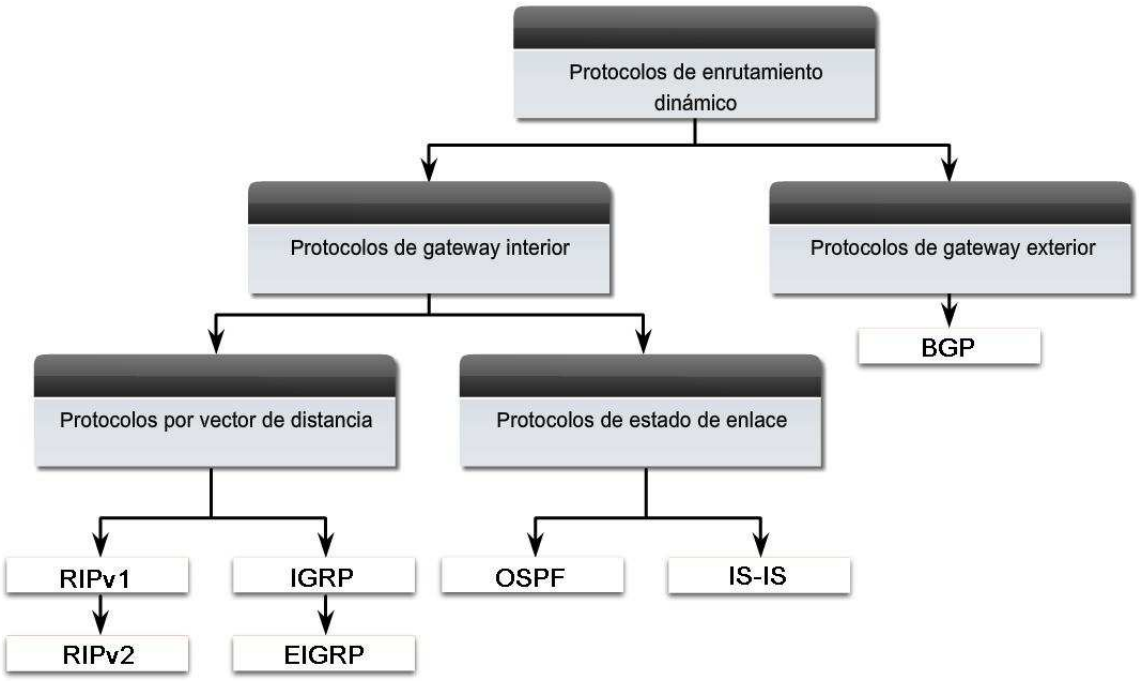

Figura 2.1 Clasificación de los protocolos de enrutamiento dinámicos.[24]

2.6.3.1 Protocolos de enrutamiento para IPv6[25]

Los siguientes protocolos de enrutamiento están definidos para IPv6:

- RIPng para IPv6
- OSPF para IPv6
- IS-IS para IPv6
- BGP-4
- IDRPv2 (Inter Domain Routing Protocol version 2)

[24] CCNA Exploration, Routing Protocols and Concepts. Cisco Networking Academy 2007.

[25] DAVIES Jhosep. Understanding IPv6, Washington USA, página 125.

2.6.3.1.1 RIPng para IPv6

Es un protocolo de vector distancia el cual es una adaptación del protocolo RIPv2. RIPng para IPv6 tiene una estructura simple de paquetes y usa el puerto UDP 521 para anunciar periódicamente sus rutas y asincrónicamente sus cambios de rutas. Tiene un número máximo de 15 saltos para alcanzar el destino y con una distancia de 16 saltos el destino es inalcanzable. Este protocolo es usado en redes pequeñas.

Cuando un router IPv6 con RIPng se inicia, esté anuncia todas la rutas en su tabla de enrutamiento en todas las interfaces; el router también envía un mensaje de solicitud general a todas las interfaces y todos los routers vecinos envían el contenido de sus tablas de enrutamiento en respuesta al mensaje y esas respuestas son las que forman la tabla inicial de enrutamiento. Las rutas aprendidas tienen un tiempo de vida de 3 minutos antes de ser eliminadas de la tabla.

Después de la inicialización RIPng anuncia periódicamente cada 30 segundos las rutas en su tabla a través de cada interfaz. El conjunto de rutas que están siendo anunciadas depende de si el router IPv6 está aplicando la regla del horizonte dividido u horizonte dividido con envenenamiento reverso.

La tolerancia a fallas se basa en el tiempo en el que RIPng aprende las rutas. Si un cambio ocurre en la topología de red los routers IPv6 con RIPng pueden enviar actualizaciones instantáneas del enrutamiento en lugar de esperar un anuncio previo.

2.6.3.1.2 OSPF para IPv6

Este protocolo está diseñado para ejecutarse como un protocolo de enrutamiento para un único sistema autónomo. OSPF para IPv6 es una adaptación del protocolo OSPFv2 para IPv4.

El costo de OSPF para cada enlace es un número único, el cual es asignado por el administrador de la red y puede incluir factores como retraso, ancho de banda y costo monetario. El costo acumulado en los segmentos de red debe ser inferior a 65535. Los mensajes de OSPF se envían como PDUs de capa superior.

Este protocolo de estado de enlace para IPv6 presenta los siguientes cambios en relación a la versión 2:

- La estructura de los paquetes OSPF ha sido modificada para eliminar las dependencias del direccionamiento en IPv4.
- Nuevas LSAs son definidas para los prefijos y direcciones en IPv6.
- OSPF se ejecuta en cada enlace en lugar de ejecutarse en cada subred.
- OSPF ya no proporciona autenticación, en su lugar OSPF se basa en la cabecera y el trailer del mensaje para realizar las tareas de autenticación.

Cada router tiene su LSA que describe su estado actual. La LSA de cada router OSPF se propaga de manera eficiente en toda la red a través de las relaciones lógicas entre los vecinos, llamadas también adyacencias; cuando la propagación de todas las LSAs se ha completado se puede decir que la red OSPF ha convergido.

Basado en la colección de LSAs de conocidos como base de datos del estado del enlace (LSDB) OSPF calcula el camino de menor costo para cada ruta y esos caminos se convierten en rutas en la tabla IPv6 de enrutamiento. Para reducir el tamaño de los LSDBs, OSPF permite la creación de zonas. Un área OSPF es la agrupación de segmentos de redes contiguos. En todas las redes OSPF debe haber por lo menos un área llamada el área de backbone.

OSPF permite la sumarización o agregación de la información de enrutamiento en los límites de un área de OSPF la cual se conoce como zona del router de frontera (ABR).

2.6.3.1.3 IS-IS para IPv6

También conocido como doble IS, es un protocolo de enrutamiento de estado de enlace muy similar a OSPF. IS-IS soporta IPv4 y CLNP (Conecctionless Red Protocol). IS-IS permite dos niveles de escala jerárquica, mientras que OSPF solo permite una.

2.6.3.1.4 BGP-4

A diferencia de RIPng y OSPF para IPv6 que son protocolos que se usan en un sistema autónomo, BGP-4 está diseñado para el intercambio de información entre sistemas autónomos. La información de enrutamiento de BGP-4 es usada para crear un árbol lógico que describe a las conexiones entre los diferentes sistemas autónomos. La información del árbol se la utiliza para la creación de rutas libres de lazos en las tablas de los routers. Los mensajes de BGP-4 son enviados por el puerto TCP 179.

Este protocolo se ha definido para ser independiente de los elementos para los cuales la información de enrutamiento está siendo propagada.

2.6.3.1.5 IDRPv2

Fue creado originalmente para CLNP y al igual BGP-4 este protocolo fue también diseñado para permitir la comunicación entre distintos sistemas autónomos, conocidos como dominios de enrutamiento en IDRP.

IDRPv2 es un mejor protocolo de enrutamiento que BGP-4 ya que en lugar de utilizar identificadores para los sistemas autónomos, en los dominios de enrutamiento IDRP se los identifica mediante un prefijo IPv6; además los dominios de enrutamiento pueden agruparse en confederaciones de enrutamiento las cuales también son identificadas por el prefijo, para crear una estructura jerárquica y así resumir el enrutamiento.

2.7 RESOLUCIÓN DE NOMBRES EN IPV6

En IPv6 son más importantes los nombres que se utilizan para hacer referencia a los recursos de red que las direcciones, ya que si en IPv4 resultaba muy complicado recordar direcciones de 32 bits, ahora en IPv6 las direcciones son de hasta 32 dígitos hexadecimales y no es razonable que los usuarios finales tengan que recordar tal dirección para intentar acceder a los recursos de la red; por lo tanto, el soporte para la resolución de nombres se constituye en un componente muy importante en IPv6.

2.7.1 RESOLUCIÓN DE DIRECCIÓN A NOMBRES

La norma 1886 de la RFC define un nuevo tipo de registro de recursos para DNS llamado “AAAA” el cual se utiliza para resolver el nombre de un dominio completo para una dirección IPv6.

Estos registros son llamados “AAAA” porque las direcciones IPv6 de 128 bits de longitud son cuatro veces más grandes que las direcciones de 32 bits que se usan en IPv4. Un registro DNS tiene la siguiente estructura: Nombre, Tipo de Registro y Dirección, donde el nombre es el nombre del dominio completo, el tipo de registro corresponde al registro que se está usando sea en IPv4 (A) o IPv6 (AAAA) y la dirección es la dirección que está asociada con el nombre. La Tabla 2.4 muestra un ejemplo de la estructura de un registro DNS para IPv6.

Nombre	Tipo de registro	Dirección
host1.microsoft.com	AAAA	FEC0::1:2AA:FF:FE3F:2A1C

Tabla 2.4 Estructura del registro de DNS para IPv6.

En DNS la asignación de una dirección IP a un nombre a menudo se denomina mapeo reverso y la norma RFC 1886 describe el IP6.INT, un nuevo dominio creado para responder inversamente a consultas de direcciones IPv6; a este dominio también se lo denomina como puntero (PTR).

Desde que DNS usa siempre un nombre como una etiqueta, las direcciones IP se convierten en un pseudo nombre con un domino de nivel superior. En IPv4 el dominio de nivel superior es “in-addr.arpa” y la dirección está escrita en cuatro etiquetas en orden inverso; por ejemplo, si el mapeo inverso para la dirección 192.0.1.2 es host1.example.org entonces el registro DNS es como se describe en la Tabla 2.5:

Nombre	Tipo de registro	Valor
2.1.0.192.in-addr.arpa	PTR	host1.ejemplo.org

Tabla 2.5 Ejemplo de PTR en IPv4.

El nombre en la parte izquierda del registro se construye invirtiendo la dirección IPv4 y añadiendo puntos entre cada dígito decimal y además añadiendo el sufijo “in-addr.arpa”. La parte derecha del registro es el nombre del host asociado con la dirección IP.

IPv6 usa el mismo registro PTR pero construye la parte izquierda del registro añadiendo puntos entre cada dígito hexadecimal de la dirección IPv6 totalmente ampliada y con un diferente y alto nivel de nombre de dominios (“ip6.arpa”). Por

ejemplo, si el mapeo inverso para la dirección 3FFE:B00:0:1::1 es host2.ejemplo.org, entonces el registro DNS es como se describe en la Tabla 2.6:

Nombre	Tipo de registro	Valor
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.b.0.e.f.f.3.ip6.arpa	PTR	host2.ejemplo.org

Tabla 2.6 Ejemplo de PTR en IPv6.

En la parte izquierda del registro de la dirección IPv6 (3FFE:B00:0:1::1) es plenamente ampliada con todos los ceros, y luego es invertido y se insertan los puntos entre cada dígito hexadecimal. Si una persona tuviera que manualmente escribir esta larga cadena en un teclado la posibilidad de cometer errores sería muy alta; por esta razón las herramientas de configuración e interfaces de usuario son necesarias para gestionar IPv6 en DNS.

El mapeo inverso (PTR) es usado mucho menos que el mapeo normal (A ó AAAA). El mapeo inverso es principalmente usado en los servidores para obtener un nombre que corresponde a una dirección IP que quiere iniciar una comunicación; esto ayuda en la recopilación de estadísticas o búsqueda de problemas y a veces ayuda en la seguridad básica.

2.7.2 TRANSPORTE[26]

DNS es una base de datos distribuída que contiene datos tanto para IPv4 como para IPv6; el mecanismo de transporte para el envío de preguntas y respuestas es independiente del tipo de datos solicitado. Por ejemplo uno podría usar IPv4 para transportar peticiones DNS en IPv6, en este caso los datos en la base de datos son direcciones IPv6 pero el mecanismo de transporte es IPv4; sin embargo, el transporte está relacionado con la consulta que se hace ya que no tiene sentido pedir una dirección IPv6 si el nodo no puede utilizar IPv6.

[26] BLANCHET Marc. Migrating to Ipv6, Quebec Canada, página 141.

Para la resolución del camino sobre un transporte IPv6, el servidor local DNS, el servidor raíz y todos los servidores principales para todos los niveles de nombres de dominio deben tener direcciones y transporte para IPv6. Desde la perspectiva del cliente solo la conexión al servidor DNS local necesita ser IPv6, todas las demás conexiones pueden estar usando como transporte IPv4.

2.8 SEGURIDAD EN IPV6

Dentro del campo de redes y más aún en las redes públicas hay un factor que se considera muy importante, y es la seguridad la cual debe aplicarse en cada componente de una red y en cada sistema; pero al hablar de seguridad se habla de un manejo de riesgos.

Existen algunas soluciones de seguridad en IPv6 las cuales se las indica en la Figura 2.2 así como también las capas y en donde trabajan.

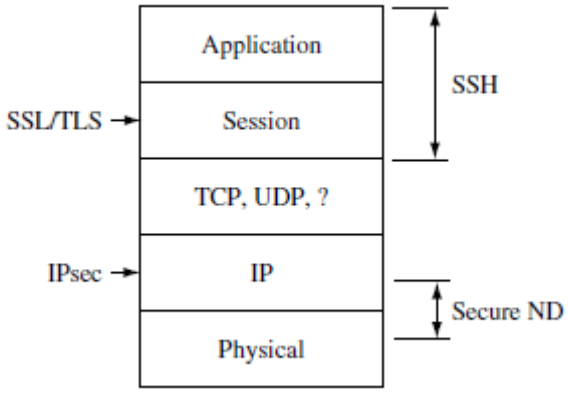

Figura 2.2 Soluciones de seguridad[27].

2.8.1 IPSEC

IPsec se definió como una extensión de las especificaciones de IPv4 pero fue diseñado de tal forma que es independiente del protocolo IP. Actualmente IPsec

[27] http://www.proprofs.com/mwiki/images/c/ca/Ipsec.

está ampliamente desarrollado en IPv4 como un método para conectar múltiple sitios remotos para la creación de una VPN a través del Internet.

En IPv6 los protocolos relacionados con IPsec son requisitos obligatorios para los nodos IPv6; este requisito no solo acelera el desarrollo de IPsec para la creación de VPNs sino también para fomentar la seguridad de las comunicaciones entre los nodos IPv6. IPsec tiene dos modos de encapsulación: transporte y túneles.

2.8.1.1 Transporte y Túnel en IPsec[28]

La Figura 2.3 muestra a los nodos N1 y N2 estableciendo una comunicación segura usando IPsec ilustrando así el modo de transporte en donde la seguridad va de extremo a extremo.

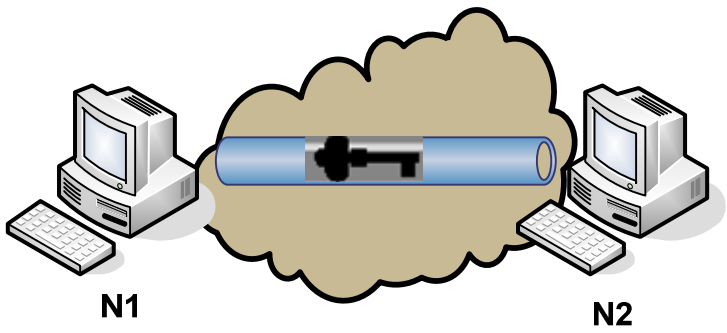

Figura 2.3 Modo trasporte de IPsec entre dos nodos.

La Figura 2.4 muestra el modo Túnel donde N1 establece una conexión IP segura con una red privada virtual (VPN) del servidor, no con N2. N1 encapsula su tráfico a N2 en una conexión IP segura hacia el servidor VPN el cual desencapsula el tráfico y lo envía hacia N2 quien recibe un tráfico no seguro de N1. Un túnel es construido desde N1 hasta el servidor VPN.

[28] BLANCHET Marc. Migrating to Ipv6, Quebec Canada, página 233.

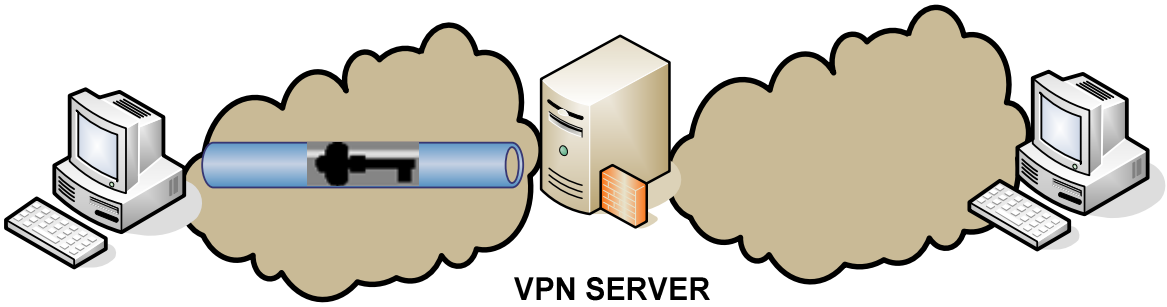

Figura 2.4 Modo Túnel de IPsec desde un nodo hacia un servidor VPN.

Los modos Túnel y Transporte se pueden combinar de modo que la comunicación entre los nodos sea más segura. La Figura 2.5 muestra el modo túnel entre N1 y el servidor VPN y el modo transporte entre el servidor VPN y N2 haciendo esto que cada paquete que viene de N1 tenga dos diferentes encapsulaciones para IPsec; una para el servidor VPN y otra para N2.

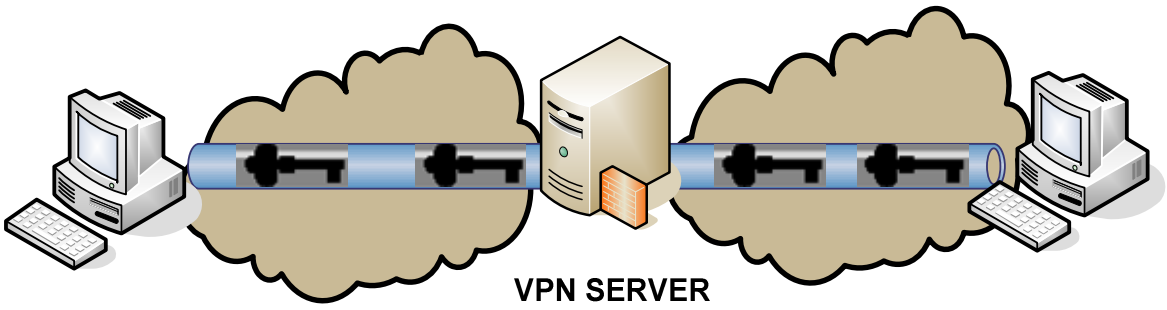

Figura 2.5 Modos IPsec Túnel y Transporte usados simultáneamente.

2.8.1.2 Asociación de Seguridad

Para cada par de dispositivos finales que están usando IPsec, una asociación de seguridad es establecida antes de un envío garantizado de paquetes. Toda esta asociación se trata de claves y algoritmos criptográficos utilizando como protocolo

el Intercambio de Claves de Internet (IKE); este protocolo usa las direcciones IP para identificar las claves utilizadas en el intercambio, lo cual hace que IKE este pendiente del protocolo IP y de las direcciones.

2.8.2 SERVICIOS DE SEGURIDAD

IPsec tiene dos servicios de seguridad: AH (Autentication Header) y ESP (Encapsulation Security Payload).

2.8.2.1 AH

Authentication Header provee:

- Integridad en todo el paquete
- Autenticación de la fuente
- Protección de la repetición de los paquetes

No todos los campos de un paquete IPv6 son protegidos AH; por ejemplo, campos como límite de saltos, clase de tráfico, etiqueta de flujo y límite de saltos no son protegidos ya que los campos en la cabecera que cambian de valores de forma imprevista a lo largo del camino no se pueden proteger. El campo que identifica la clase de tráfico puede ser cambiado a través de routers que usan servicios diferenciados (diffserv), el campo de control de flujo (flow Label) puede ser cambiado entre diferentes dominios de QoS, el campo que indica el número máximo de saltos es decrementado por cada router a lo lago del camino. Los otros campos de la cabecera IPv6 están protegidos por AH. La protección es hecha mediante el cifrado de la suma de comprobación de la información que viaja por el camino; esta suma de comprobación así como la asociación de seguridad de la información son almacenadas en la extensión de la cabecera AH, la cual es identificada por el valor 51 en el siguiente campo de cabecera.

2.8.2.2 ESP

Encapsulation Security Payload provee:

- Confidencialidad
- Integridad del paquete interno
- Autenticación de la fuente
- Protección contra la anti-reproducción de los paquetes

Comprada con AH, ESP añade confidencialidad (cifrado). La extensión de la cabecera de ESP es identificada por el valor 51 en la cabecera del siguiente campo. Ningún campo de la cabecera IPv6 está protegida sino solamente el contenido de la carga útil incluyendo al protocolo de transporte; si ESP utiliza servicio de confidencialidad entonces la carga útil es encriptada.

2.8.2.3 IPsec e IPv6

Dadas las amenazas actuales de las redes y la necesidad de seguridad en la capa IP IPsec llegó a ser popular. Sin embargo, IPv4 NAPT(Netwok Address Port Translation) causa muchas complicaciones para los usuarios, de modo que IPsec no se usa frecuentemente debido a las complicaciones de IPv4 NAPT. Por otro lado, en IPv6 IPsec debe ser usado de forma obligatoria por todos los nodos y no tiene que hacer frente a las limitaciones NATP necesarias para IPv4, además, IPsec es más fácil de implementar sobre IPv6 y ayudará a asegurar el Internet, las redes de las empresas y el hogar.

Dado que es obligatoria la implementación de IPsec en IPv6 esto permitirá a los desarrolladores el diseño de nuevas aplicaciones y protocolos de IPsec utilizando la infraestructura existente para IPv6 y así las aplicaciones y protocolos serán más seguros desde el comienzo.

2.8.3 FILTROS Y FIREWALLS

Las políticas de seguridad deben ser aplicadas para el tráfico IPv6 por las mismas razones por las cuales se las aplica en IPv4; sin embrago, existen algunas consideraciones que se deben tomar en cuenta para IPv6.

2.8.3.1 Filtrado ICMP

Los nodos IPv6 dependen del “Path MTU Discovery” para obtener el máximo rendimiento en la conexión. Path MTU Discovery se basa en el paquete ICMP que contiene el mensaje de error más grande el mismo que es enviado desde los routers en la red. Algunos sitios implementan sus políticas de seguridad en IPv4 previniendo que los mensajes ICMP pasen por sus firewalls; para IPv6 los mensajes ICMP específicos deberían pasar a través de los firewalls para que los nodos trabajen correctamente.

2.8.3.2 NAPT

Oculta la identidad de los nodos en IPv4 que están en la red privada; si la política de seguridad se aplica después del proceso que es realizado por NAPT no hay forma de permitir o denegar el tráfico de los nodos que se encuentran detrás de NAPT pues no son reconocibles a partir de la dirección de origen y sus puertos; además, la solución de problemas de tráfico cuando se usa NAPT es muy difícil debido a que para buscar algún fallo se debe realizar una sincronización de todos los eventos dentro del tráfico y aún se vuelve más complicado si dentro del camino existen puntos en los cuales NAPT está siendo aplicado. Sin el uso de NAPT en IPv6 la aplicación de políticas de seguridad y la solución de problemas es más fácil ya que los campos de la cabecera del paquete no cambian en el camino.

2.8.4 DIRECCIONES TEMPORALES[29]

En IP, cuando un nodo se traslada a otro sitio recibe una nueva dirección. Este cambio de dirección hace que sea difícil para algunos servidores localizar al nodo cuando se mueve; por eso los sitios Web usan cookies para realizar un seguimiento de los usuarios sin importar la dirección IP que puedan tener.

Cuando un nodo IPv6 autoconfigurado se mueve, el prefijo cambia pero el identificador de interfaz sigue siendo el mismo, esto debido a que los nodos usan un mecanismo de autoconfiguración en la red y el identificador de la interfaz (como parte de la dirección IPv6) sigue basándose en la misma dirección de Capa 2. Esto plantea algunas dudas sobre la privacidad ya que el nodo IPv6 puede ser fácilmente seguido aún cuando este moviéndose de lugar. Una solución temporal puede ser utilizar identificadores de interfaces que sean modificados a menudo para así de esta forma lograr privacidad. La recomendación RFC 3041 define una manera de garantizar la aleatoriedad de los cambios de la dirección lógica de modo que no se puedan adivinar las nuevas direcciones en base a las direcciones antiguas.

Cuando se genera una nueva dirección temporal, la dirección antigua si bien ya no es correcta puede recibir aún paquetes de conexiones ya establecidas.

Las tarjetas de red son frecuentemente diseñadas con unos pocos números de registros para el procesamiento de las direcciones multicast. Reducir al mínimo el número de grupos multicast a los cuales escuchar ayuda a que el procesamiento multicast sea directamente realizado por la tarjeta. El uso de direcciones aleatorias generadas temporalmente rompe esta optimización ya que los últimos 24 bits serán diferentes y así cada dirección temporal generará una nueva solicitud a una dirección multicast de los nodos para que sea escuchada.

---

[29] BLANCHET Marc. Migrating to Ipv6, Quebec Canada, página 245.

2.8.5 SEGURIDAD EN UN ENLACE IPV6[30]

Cuando un intruso tiene acceso físico a un enlace, muchas herramientas se pueden utilizar para realizar ataques, las cuales representan una amenaza para la seguridad. Muchas amenazas han sido identificadas de acuerdo a la interacción que se tiene en un enlace local y estas no son nuevas en IPv6 ya que la mayoría son realizadas en IPv4 a través de mensajes ARP, ICMP y DHCP.

Los routers deben ser confiables para los nodos. Antes de la conexión el nodo está pre-configurado con una lista de elementos que son de confianza identificados por claves públicas y nombres, los cuales son autorizados para emitir certificados para los routers. Cuando un nodo se conecta a un enlace, este envía un mensaje de solicitud a todos los routers en el enlace pidiendo todos los certificados relacionados con la lista pre-configurada. El router responde enviando los certificados que fueron firmados por la lista de confianza especificada en la solicitud. El nodo envía al router una solicitud a la que el router responde con un mensaje RA (router advertisement) firmado por uno de los certificados. El nodo puede entonces validar los mensajes del router y así es como un nodo puede confiar en los mensajes del router.

IPv6 es un nuevo protocolo y es implementado con un nuevo código, pero el hecho de que sea un nuevo código implica también que lleva nuevos defectos de seguridad también; sin embargo, IPv6 corrige muchas cuestiones de seguridad relacionadas con IPv4.

Algunos nuevos ataques han sido descubiertos y ahora algunos de los ataques antiguos con el uso de IPv6 son difíciles de lograr. Otro aspecto importante es que IPv6 no usa NAPT, lo cual ayuda al diseño de las políticas de seguridad, búsqueda de problemas y al desarrollo de servicios de seguridad como IPsec.

---

[30] BLANCHET Marc. Migrating to Ipv6, Quebec Canada, página 248.

IPv6 se está convirtiendo en un protocolo más seguro que IPv4 ya que los nuevos protocolos en la suite de protocolos de IPv6 proveen mayor confiabilidad y añaden mayor seguridad en la entrega de datos; algunos de estos nuevos protocolos no están disponibles en IPv4.

2.9 TRANSICIÓN DE IPV4 A IPV6

Siempre las transiciones de un estado a otro no son fáciles y la transición de IPv4 a IPv6 no es la excepción. Cuando se habla de transiciones de protocolos al momento de realizar los cambios necesarios y correspondientes hay que verificar que todo funcione correctamente con la instalación y configuración del nuevo protocolo. Aunque esto pueda resultar fácil en una red pequeña, el reto de realizar una transición en una organización grande se vuelve difícil. Además, dado el alcance que el Internet tiene, la transición rápida de un protocolo en todo un medio, se convierte en una tarea complicada.

Los desarrolladores de IPv6 reconocen que la transición de IPv4 a IPv6 tomará años y que habrá organizaciones que aún sigan usando IPv4 indefinidamente, así que se debe tomar en cuenta que es necesario una coexistencia de IPv4 e IPv6.

Los creadores de IPv6 mediante la recomendación RFC 1752 definen los siguientes criterios de transición:

- Los hosts IPv4 pueden ser actualizados en cualquier tiempo, independientemente de la actualización de otros hosts u otros routers.
- Los nuevos hosts que solo usan IPv6 pueden ser añadidos en cualquier momento sin depender de la infraestructura de otros hosts o routers.
- Los hosts IPv4 que tiene instalado IPv6, pueden continuar usando su dirección IPv4 y no necesitan de direcciones adicionales.
- Se requiere de una pequeña preparación para actualizar los nodos IPv4 a IPv6.

La inherente falta de dependencias entre los hosts IPv4 e IPv6, la infraestructura de enrutamiento de IPv4 e IPv6 requiere de una serie de mecanismos los cuales ayudarán a mantener una convivencia perfecta.

Para que la coexistencia pueda lograrse exitosamente, los nodos deben usar la infraestructura de IPv4, IPv6 o una combinación de ambas ya que la verdadera migración se logra cuando todos los nodos IPv4 se convierten solo en nodos IPv6. Sin embargo actualmente en la práctica, la migración se logra cuando la mayoría de los nodos IPv4 llegan a convertirse en nodos que usan tanto IPv4 como IPv6.

Para la coexistencia con infraestructuras que usan IPv4 y proveer una eventual y paulatina migración a IPv6, se utilizan los siguientes mecanismos:

1. Capa dual IP
2. Túneles IPv6 sobre IPv4
3. Infraestructura DNS

2.9.1 CAPA DUAL IP[31]

La capa dual IP es una implementación de la suite de protocolos TCP/IP que incluye IPv6 e IPv4; es un mecanismo usado por ambos protocolos de modo que la comunicación entre nodos IPv4/IPv6 pueda suceder. Una capa dual IP contiene una única implementación de protocolos usados en comunicaciones host-a-host tales como TCP y UDP.

Todos los protocolos de capas superiores en una implementación de capa dual IP pueden comunicarse mediante IPv4, IPv6 o por un túnel IPv6 sobre IPv4.

[31] DAVIES Jhosep. Understanding IPv6, Washington USA, página 250.

La Figura 2.6 muestra la arquitectura de la capa dual IP; a veces esta arquitectura se divide en protocolos TCP/UDP por separado para IPv6 e IPv4 ya que en algunos sistemas esta arquitectura no funciona como capa dual.

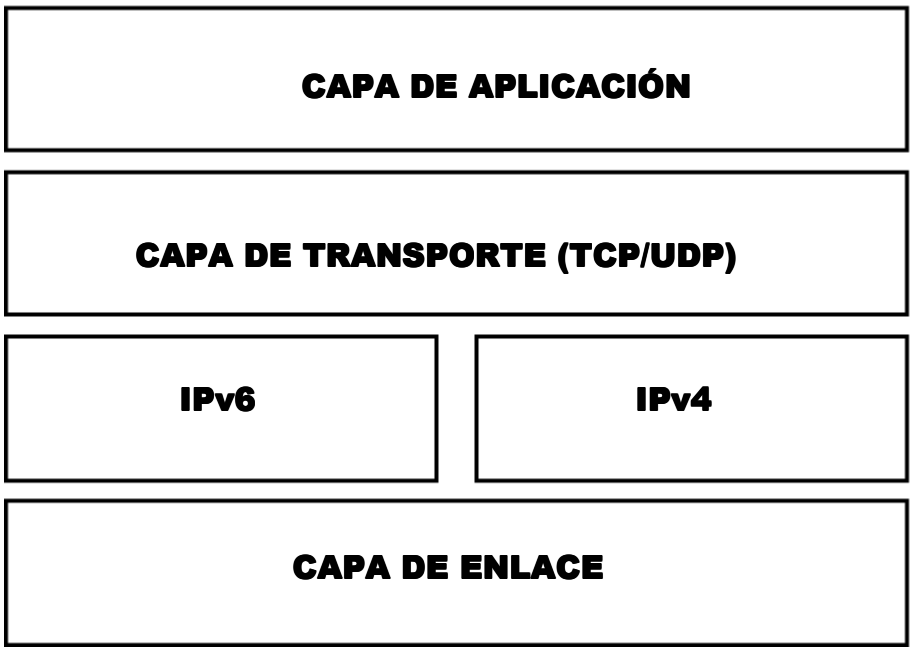

Figura 2.6 Arquitectura de la capa dual IP.

2.9.2 TÚNELES IPV6 SOBRE IPV4[32]

Tunneling (túnel IPv4/IPv6) es una técnica en donde se encapsulan los paquetes IPv6 en una cabecera IPv4 de modo que los paquetes IPv6 puedan ser enviados sobre una infraestructura IPv4. En la cabecera IPv4 el valor del campo de protocolo es configurado con el valor de 41 para indicar que un paquete IPv6 ha sido encapsulado y los campos de origen y destino son configurados con direcciones IPv4 en los puntos finales del túnel. Los extremos del túnel son configurados manualmente ya sea como parte de la interfaz del túnel o se deriva automáticamente de la interfaz que está enviando los paquetes, la dirección del

[32] DAVIES Jhosep. Understanding IPv6, Washington USA, página 255.

próximo salto de la ruta correspondiente o de las direcciones IPv6 de origen o destino en la cabecera IPv6.

La Figura 2.7 muestra como se encapsula un paquete IPv6 en uno IPv4.

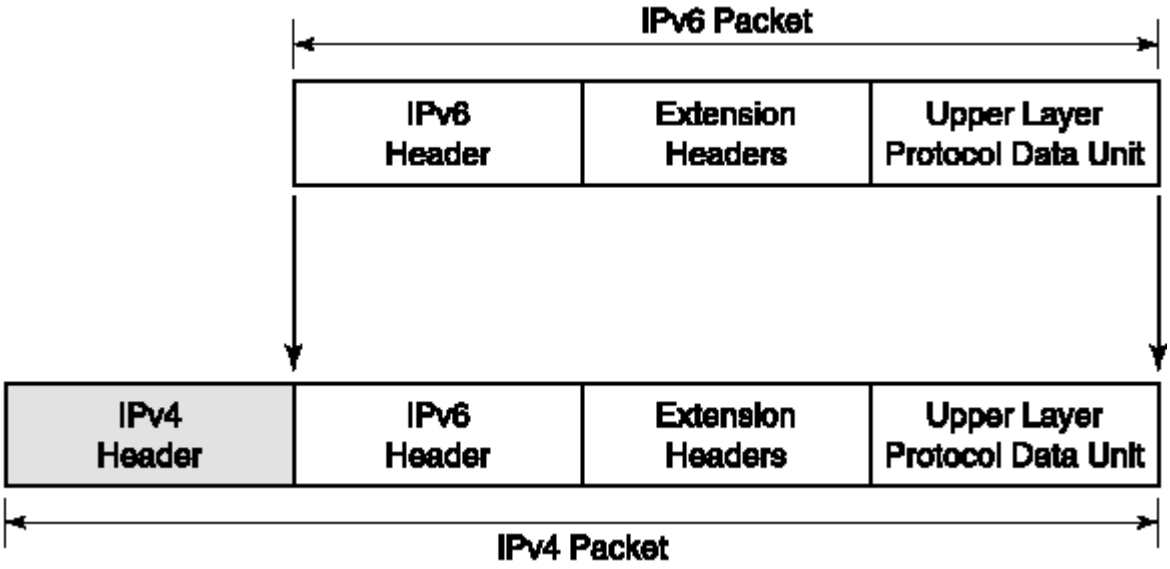

Figura 2.7 Arquitectura de la encapsulación IPv6 sobre IPv4[33]

Hay casos en los que si la ruta de acceso de IPv4 no es almacenada en cada túnel, el paquete IPv4 deberá ser fragmentado por un router intermedio; en IPv6 se debe enviar el paquete con una bandera “No Fragmentar” en la cabecera IPv4.

Las siguientes configuraciones son usadas en los túneles: router-a-router, host-a-router y router a host, y host-a-host.

2.9.2.1 Router-a-router[34]

En este tipo de configuración las dos infraestructuras IP están conectadas por dos routers IPv4/IPv6 sobre una infraestructura IPv4. Los extremos de túnel abarcan un enlace lógico entre el origen y el destino. El túnel IPv6 sobre IPv4 entre los dos

[33] DAVIES Jhosep. Understanding IPv6, Washington USA, página 255.

[34] DAVIES Jhosep. Understanding IPv6, Washington USA, página 260.

routers actúa como un camino de un solo salto. La Figura 2.8 muestra un ejemplo del túnel router-a-router.

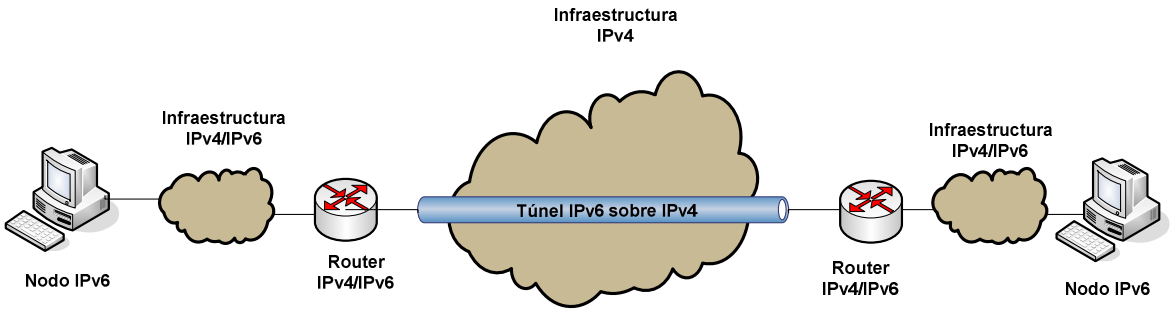

Figura 2.8 Túnel Router-a-router.

Esta configuración se puede usar en:

- Un laboratorio de prueba IPv6 cuyos túneles crucen una infraestructura IPv4 para alcanzar el Internet IPv6.
- Dos dominios de enrutamiento IPv6 en donde el túnel cruce por el Internet IPv4.
- Un router 6to4 cuyo túnel cruce el Internet IPv4 para llegar a otro router 6to4.

2.9.2.2 Host-a-router y router a host[35]

En esta configuración un nodo IPv6/IPv4 que reside dentro de un infraestructura IPv4 crea un túnel IPv6 sobre IPv4 para llegar a un router IPv6/IPv4. Los extremos del túnel abarcan el primer segmento de la ruta entre el origen y el destino de los nodos. El túnel IPv6 sobre IPv4 entre el nodo IPv6/Ipv4 y el router IPv6/IPv4 actúa como una ruta de un solo salto.

---

[35] DAVIES Jhosep. Understanding IPv6, Washington USA, página 265.

En el nodo IPv4/IPv6 y en el router IPv6/IPv4 mediante un representación de una interfaz en el túnel se crean rutas para el nodo y para el router, las cuales se calculan utilizando la interfaz del túnel. La Figura 2.9 muestra el túnel host-a-router y router-a-host.

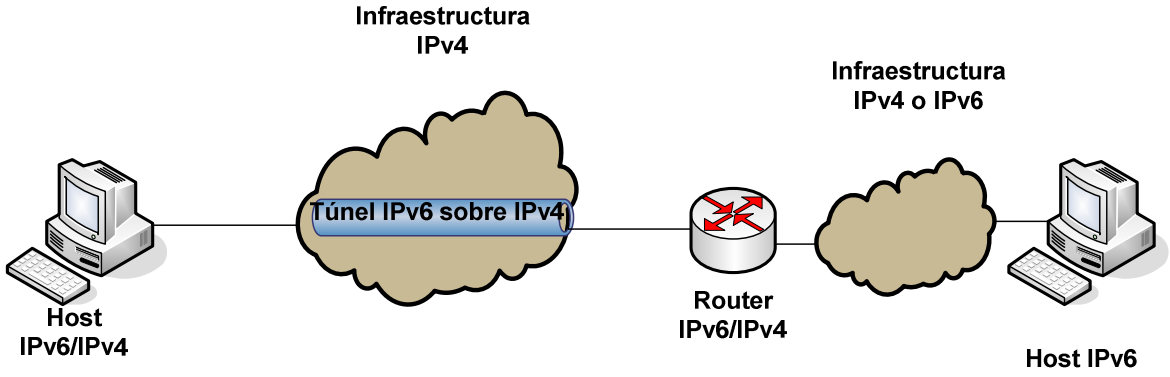

Figura 2.9 Túnel host-a-router y router-a-host.

Esta configuración se puede usar en:

- Un host IPv6/IPv4, en donde el túnel cruza la infraestructura IPv4 de una organización para alcanzar el Internet IPv6.
- Un host ISATAP, donde el túnel cruza una red IPv4 por un router ISATAP para alcanzar el Internet IPv4, otra red IPv4 u otra red IPv6.
- Un router ISATAP, donde el túnel cruza por una red IPv4 para llegar a un host ISATAP.

2.9.2.3 Host-a-host[36]

En esta configuración un nodo IPv6/IPv4 que reside en una infraestructura IPv6 crea un túnel IPv6 sobre IPv4 para llegar a otro nodo que reside con la misma infraestructura. Los routers pueden estar presentes para indicar que el nodo de

[36] DAVIES Jhosep. Understanding IPv6, Washington USA, página 266.

destino está en la misma subred de la infraestructura IPv4. Este tipo de túnel puede ser usando en:

- Hosts IPv6/IPv4 que usan direcciones ISATAP para encaminarse mediante un túnel a la infraestructura IPv4 de una organización.
- Hosts IPv6/IPv4 que usan direcciones IPv4 para encaminarse mediante un tunel a la infraestructura IPv4 de una organización.

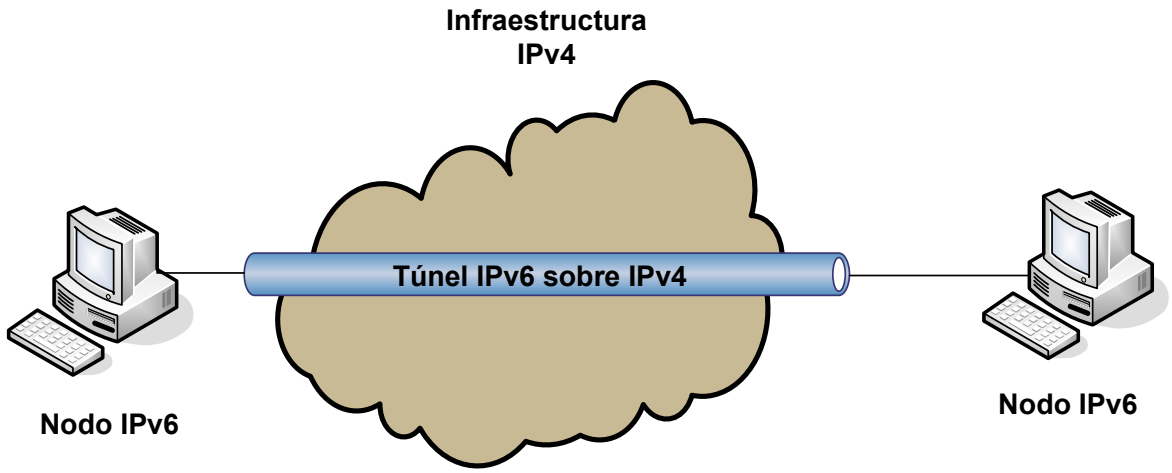

Figura 2.10 Túnel Host-a-host

2.9.3 TIPOS DE TUNELES[37]

Existen dos tipos de túneles: túneles configurados y automáticos.

Los túneles configurados o estáticos requieren configuración manual en los extremos del túnel; las direcciones IPv4 de los extremos del túnel no están codificadas en direcciones IPv6 de origen o destino. Este tipo de túnel se usa en las configuraciones router-a-router y host-a-router, y las configuraciones de la interfaz del túnel deben ser especificadas manualmente a lo largo del túnel con rutas estáticas. Para la configuración de túneles estáticos se debe cumplir que

[37] DAVIES Jhosep. Understanding IPv6, Washington USA, página 268.

ambos routers sean dual-stack y que los routers tengan direcciones IPv4 que sean alcanzables; además esta configuración se aplica cuando se necesita un número muy pequeño de túneles y NAT no está presente en el camino.

Los túneles automáticos no necesitan de configuración manual; los puntos finales del túnel son determinados por el uso de interfaces lógicas en el túnel, rutas, y direcciones IPv6 de origen y destino. Estos son usados cuando las direcciones compatibles con IPv4 (::w.x.y.z donde w.x.y.z es una dirección IP pública) se utilizan. Los túneles automáticos también son usados en la configuración de túnel host-a-host entre dos host IPv4/IPv6 usando direcciones IPv4 compatibles.

2.9.4 6OVER4[38]

Este mecanismo de transición conocido también como túnel multicast IPv4, usa la capacidad multicast de una red IPv4 para permitir que la red IPv4 actúe como una subred virtual para los hosts y routers IPv6. 6over4 trata a la infraestructura IPv4 como un enlace único con capacidad para multicast. El proceso ND trabaja como lo hace sobre cualquier enlace físico multicast. La Figura 2.11 muestra la arquitectura de 6over4.

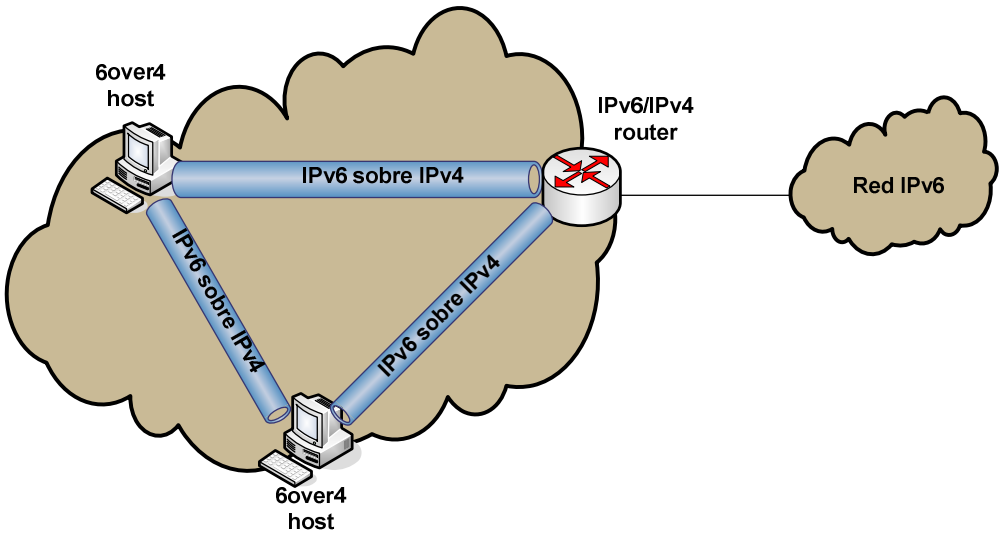

Figura 2.11 Arquitectura de 6over4.

[38] DAVIES Jhosep. Understanding IPv6, Washington USA, página 275.

Los hosts 6over4 usan un prefijo válido de 64 bits para las direcciones unicast y el identificador de interfaz :WWXX:YYZZ donde WWXX:YYZZ representa a una dirección IPv4 (w.x.y.z) asignada al host. Por defecto los hosts 6over4 son configurados con direcciones de enlace local FE80::WWXX:YYZZ en cada interfaz 6over4.

Para facilitar las comunicaciones IPv6 multicast sobre una infraestructura multicast IPv4. La recomendación 2559 define el siguiente método para traducir una dirección IPv6 multicast en una dirección IPv4 multicast:
239.192.(penúltimo byte de la dirección IPv6).(último byte de la dirección IPv6)
Por ejemplo la dirección FF02::1:FF28:9C5A en IPv6 es mapeada a 239.192.156.90 en IPv4.

Ya que la infraestructura IPv4 actúa como un enlace con capacidad de multicast los hosts pueden usar mensajes NS (Neighbor Solicitation) y NA (Neighbor Advertisement) para resolver las direcciones de capa de enlace. Las direcciones IPv4 embebidas en la porción del identificador de la interfaz de las direcciones 6over4 están en los extremos del túnel. Los hosts y los routers pueden usar los mensajes RS (Router Solicitation) y RA (Router Advertisement) para los prefijos de los routers y otros parámetros.

2.9.5 6TO4[39]

Este mecanismo de transición es muy importante ya que provee una interconexión de dominios IPv6 aislados mediante túneles automáticos a través del Internet IPv4. La motivación de este método es permitir que los dominios aislados IPv6 o hosts conectados a una red IPv4 puedan comunicarse con otros dominios IPv6 o hosts con un mínimo de configuración manual; el mecanismo también permite la conexión de esos dominios al Internet IPv6 sobre el Internet IPv4.

[39] AMOSS John, MINOLI Daniel. IPv4 to IPv6 transition, New York USA, página 129.

6to4 trata al Internet IPv4 como un enlace unicast punto a punto con el fin de interconectar los dominios aislados IPv6 entre sí y con el Internet IPv6. El entunelamiento automático se logra al tener un router 6to4 en la frontera de los dominios aislados IPv6 y conectado al Internet IPv4.

El mecanismo de transición opera teniendo la dirección IPv4 de la interfaz de su router, o la dirección IPv6 asignada a un host en el respectivo dominio IPv6; por lo tanto, especificando la dirección IPv6 de un host que está usando 6to4, explícitamente se identifican el punto final del túnel IPv4 del router de frontera 6to4.

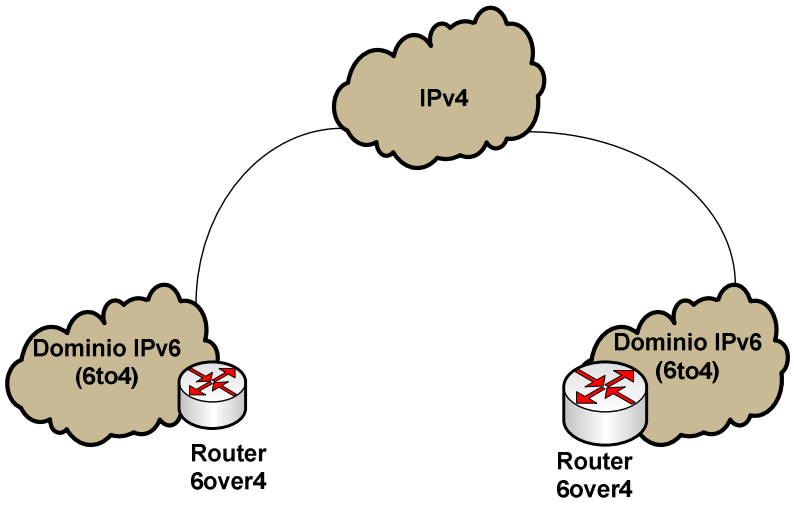

Figura 2.12 Interconexión de dominios IPv6 mediante 6to4.

Se deben considerar aspectos principales en 6to4 como:

- Cuando el router de frontera 6to4 de un dominio 6to4 está dirigiendo un paquete IPv6 a un host en otro domino 6to4, el router puede automáticamente determinar si el punto final del túnel desde una dirección IPv4 embebida es una dirección 6to4 IPv6.
- Este mecanismo conserva las direcciones IPv4 porque solo una única dirección global IPv4 es necesitada para todo un dominio IPv6.
- Este mecanismo cae en una configuración de túnel router-a-router.

Adicionalmente, para interconectar dominios 6to4, el mecanismo 6to4 permite la interconexión de esos dominios con redes nativas IPv6 a través de Bridges o relay routers, entre los sitios 6to4 y las redes IPv6. La mayoría de elementos que forman parte del mecanismo de transición 6to4 son los siguientes:

- **Host 6to4:** es un host IPv6 que está configurado con al menos una dirección 6to4 IPv6.
- **Router 6to4:** Es un router IPv4/IPv6 que soporta el uso de interfaces de túnel y es típicamente usado para direccionar tráfico 6to4 entre hosts 6to4 con un sitio y otros routers 6to4 o Bridges 6to4 sobre el Internet IPv4. Los routers 6to4 requieren un procesamiento lógico y talvez una configuración manual para realizar de forma correcta la encapsulación y la desencapsulación.
- **Relay router 6to4:** o brigde, envía tráfico entre routers 6to4 a través del Internet y hosts sobre el Internet IPv6.

2.9.5.1 Direccionamiento 6to4[40]

La recomendación RFC3056 define un método para asignar un único prefijo de dirección IPv6 para los hosts en un dominio 6to4. Para este propósito IANA ha asignado permanentemente un espacio de dirección IPv6 para 6to4 que es: 2002::/16. El resto del prefijo se obtiene añadiendo los 32 bits de las direcciones IPv4 asignadas al router 6to4. La Figura 2.12 muestra un ejemplo del direccionamiento que 6to4 realiza: el router de frontera tiene una dirección externa IPv4 192.0.2.1, el sitio IPv6 detrás del router utiliza la dirección 2002:C000:0201::/48 para identificar a toda su red. El espacio de dirección está basado en 2002:(dirección IP externa en hexadecimal)::///48, pero la dirección IPv4 externa del router es 192.0.2.1 y en hexadecimal es C000:0201. el

[40] BLANCHET Marc. Migrating to Ipv6, Quebec Canada, página 285.

mecanismo 6to4 solo necesita ser implementado en los routers de frontera, los hosts dentro del sitio IPv6 no necesitan soporte de 6to4.

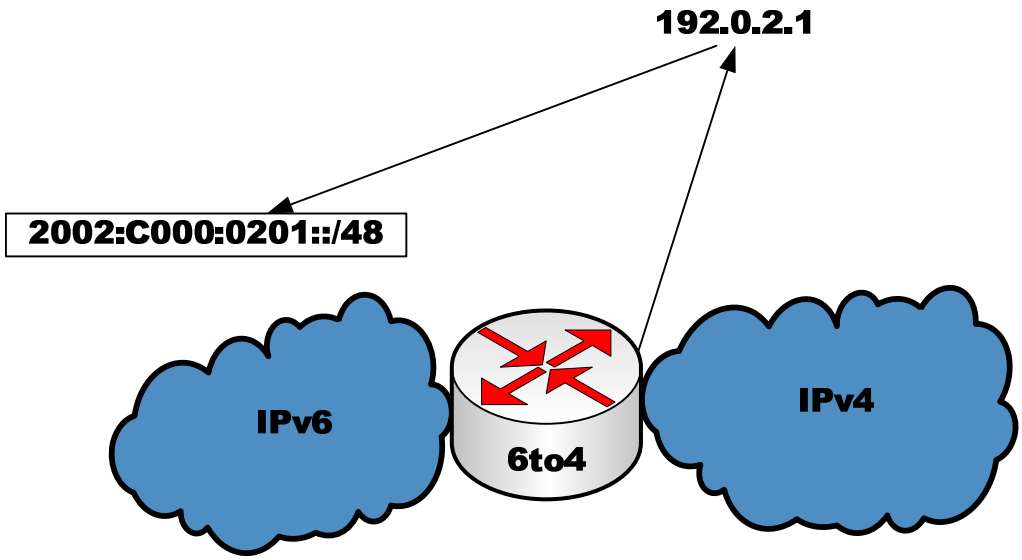

Figura 2.12 Direccionamiento 6to4 basado en un router de frontera.

En 6to4 también se tienen los procesos de tunneling (o configuraciones de túneles) vistos anteriormente los cuales son router-a-router, host-a-host y host-a-router.

2.9.6 ISATAP[41]

ISATAP (Intra Site Automatic Tunnel Addressing Protocol) es un mecanismo para automatizar la creación de túneles desde los nodos a los routers y desde los nodos a nodos dentro de un mismo sitio. ISATAP es similar a 6over4 con la diferencia de que ISATAP no asume una infraestructura multicast.

[41] BLANCHET Marc. Migrating to Ipv6, Quebec Canada, página 290.

2.9.6.1 Direccionamiento[42]

Este mecanismo incluye la dirección IPv4 del nodo en los últimos 32 bits del identificador de interface como parte de su dirección IPv6. la Figura 2.13 muestra el formato de una dirección ISATAP. Los primeros 32 bits del identificador de interface son 00:00:5E:FE los que son reservados por IANA para ISATAP.

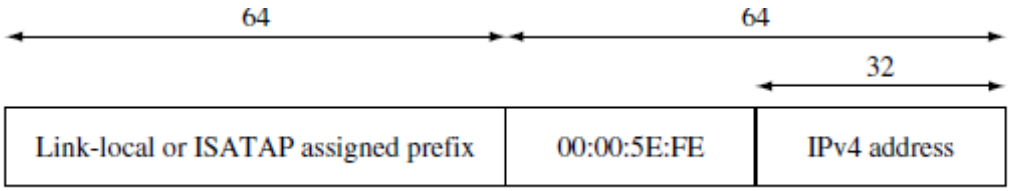

Figura 2.13 Formato de dirección ISATAP.

2.9.6.2 Tunneling[43]

ISATAP crea un enlace virtual a través de un sitio IPv4. La Figura 2.14 muestra un ejemplo de una red ISATAP, el host A, el host B y el router R1 son dual-stack y tienen habilitado ISATAP. El host C es IPv6 pero no tiene habilitado ISATAP. El administrador de la red usa la dirección 3FFE:B00:1::/48 para la red y asigna la dirección 3FFE:B000:1:2::/64 para el enlace virtual ISATAP sobre la red IPv4. El host A tiene la dirección IPv4 192.0.2.1 y crea una dirección de enlace local basándose en el formato ISATAP: FE80::5EFE:C000:0201, la misma que es calculada usando los 32 bits del identificador de interfaz ISATAP (000:5EFE) y la representación hexadecimal de su dirección IPv4 (C000:0201 – 192.0.2.1). El host B y el router R1 hacen lo mismo, respectivamente. El host A envía un mensaje RS a la dirección ISATAP del router R1 configurado estáticamente en el nodo, luego el host A recibe un mensaje RA del router R1. El host A configura su dirección global como 3FFE:B000:1:2:5EFE:C000:0201 desde que 3FFE:B000:1:2::/64 es

---

[42] AMOSS John, MINOLI Daniel. IPv4 to IPv6 transition, New York USA, página 133.

[43] BLANCHET Marc. Migrating to Ipv6, Quebec Canada, página 291

el prefijo del enlace virtual recibido del router. El router también incluye los avisos de que el mismo se está identificando como default router, el aviso de su dirección ISATAP como próximo salto. El host B hace lo mismo, respectivamente, lo cual mejora el enlace virtual con direcciones globales y un default router para alcanzar la red IPv6. Cuando el host A envía un paquete a B, lo mira en el mismo enlace de modo que A envía un mensaje NS a la dirección de B. B responde con un mensaje NA. Estos mensajes son encapsulados dentro de los paquetes IPv4 unicast.

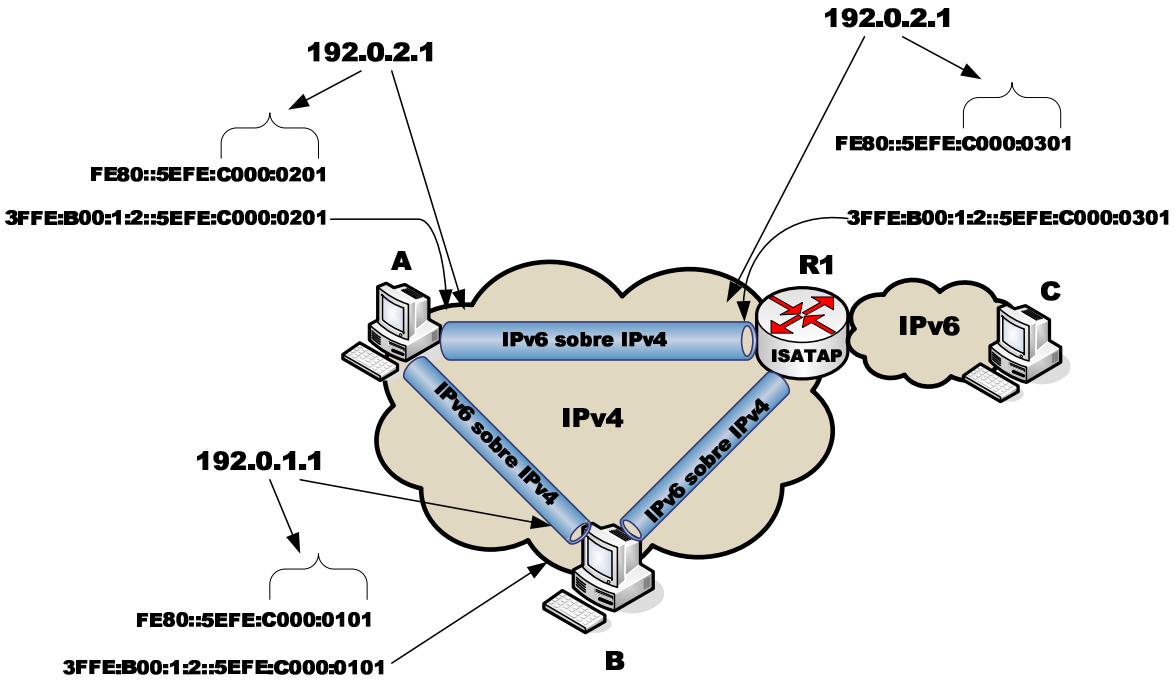

Figura 2.14 Red ISATAP.

En este capítulo se han revisado algunas diferencias entre IPv4 e IPv6 así como también varios conceptos y fundamentos de estos protocolos, además de los mecanismos de transición para la migración a IPv6. En el siguiente capítulo se procederá a realizar un plan para la migración del ISP a IPv6, teniendo en consideración los temas que en el presente capítulo se trataron.

CAPÍTULO 3

PLAN PARA LA MIGRACIÓN DE IPV4 A IPV6

CAPÍTULO 3

PLAN PARA LA MIGRACIÓN DE IPV4 A IPV6

3.1 INTRODUCCIÓN

La migración de IPv4 a IPv6 es muy compleja ya que se debe primero cumplir una serie de requerimientos para que la migración pueda verse realizada; algo que debía esperarse largo pues no se puede en poco tiempo completar todo un proceso que aún sigue en desarrollo, pero, con el avance obtenido hasta hoy en día, ha logrado resolver y facilitar muchos problemas en la comunicación sobre IPv4, además de mitigar el problema del agotamiento de las direcciones.

3.2 CONSIDERACIONES GENERALES

No cabe duda que la migración constituye un largo y complejo proceso por lo cual algunas organizaciones han emitido ciertas recomendaciones y metodologías generales (como guías especiales para los ISPs) para que este proceso se pueda dar. La recomendación RFC 4029 describe algunos pasos que los ISPs deberían considerar para una migración:

- Obtener espacio para las direcciones IPv6.
- Elaborar un plan de asignación de direcciones IPv6
- Estudiar sobre las herramientas disponibles para el manejo y monitoreo de la red.
- Actualizar a los nodos para que funcionen tanto con IPv4 como con IPv6.

- Seleccionar el apropiado protocolo de enrutamiento en IPv6 y establecer políticas de enrutamiento, las que pueden ser las mismas que las usadas en IPv4.
- Implementar algún mecanismo de transición (como por ejemplo túneles).
- Habilitar los servicios IPv6 que se requieren (DNS, QoS, etc).
- Habilitar los equipos en los sitios finales (equipos en el lado del usuario).
- Y por último convertir todos los nodos IPv4/IPv6 en nodos IPv6 solamente.

Además la RFC sugiere que una actualización de los backbones de los ISPs puede ocurrir según las adquisiciones que estos realicen, ya que algunos pueden poseer equipos y plataformas que no soporten IPv6, y en cuyo caso una actualización de los equipos es una buena decisión, pero, también se debe tener en cuenta que algunos equipos permiten actualizar su software o realizar ciertas adecuaciones para el soporte de este nuevo protocolo.

3.3 REQUIRIMIENTOS PARA LA MIGRACIÓN

La Figura 3.1 muestra un esquema de la estructura de la red principal y sus diferentes componentes.

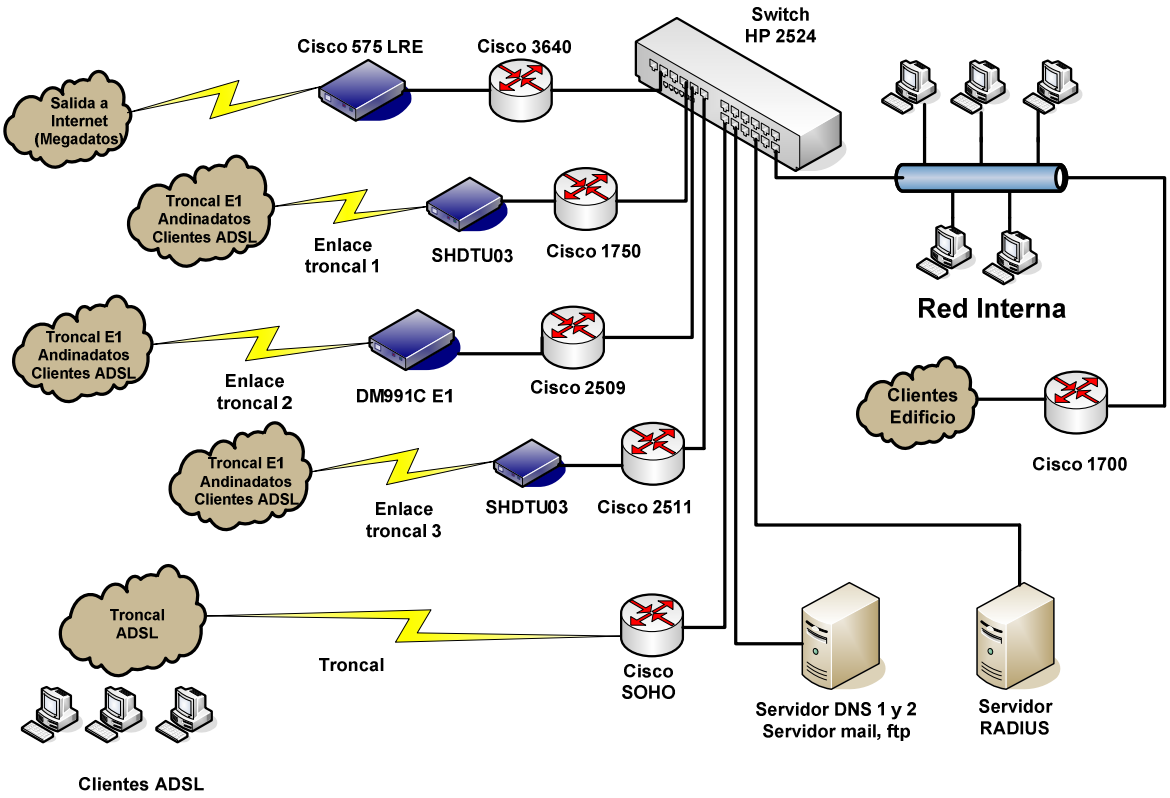

Figura 3.1 Backbone del ISP.

Todos los routers son marca Cisco y esto en parte representa una ventaja ya que Cisco, como parte activa del grupo que definió la estandarización de IPv6 y miembro fundador del "IPv6 Forum", ha implementado en sus equipos los principales mecanismos de transición a IPv6 para que puedan usarse a partir de la versión Release 12.2(2)T del IOS que es la que admite las funcionalidades de IPv6.

Los routers de los enlaces troncales: Cisco 1750, Cisco 2509, Cisco 2511 y el router de la red local Cisco 1700 deben ser necesariamente reemplazados ya que tienen versiones inferiores a la 12.2(2)T del IOS y tampoco permiten una actualización del IOS. Existen 3 modelos de routers que podrían reemplazar a los equipos que no soportan IPv6, estos son: Cisco 1841, Cisco 3825 y Cisco 2821.

De estos tres modelos, el router Cisco 2821 tiene una memoria RAM de 256MB y expandible hasta 1GB, una memoria flash de 64MB expandible hasta 256MB y también incluye un firewall IPv6, de modo que este modelo puede reemplazar a los otro routers que no soportan IPv6. El router Cisco 1841 es recomendado para empresas pequeñas empresas y podría este reemplazar al router Cisco 1700 que se encuentra en la red local del ISP y brinda servicios a los clientes del edificio, pero, dado que una migración a IPv6 puede representar el proveer nuevos servicios por parte del ISP, como las aplicaciones de VoIP por ejemplo, el router Cisco 2821 posee un mejor soporte para estas aplicaciones. El router Cisco 3825 es usado para empresas grandes y su uso se basa principalmente en las aplicaciones de VoIP, además es demasiado costoso en relación a los Cisco 1841 y 2821. Por estas razones el router Cisco 2821 sería el equipo que el proveedor puede usar para reemplazar tanto a los routers de los enlaces troncales como para el router usado en la red local. Además este equipo es recomendado por Cisco para los proveedores de Internet que estén empezando a planificar su migración a IPv6[44]

Los servidores en Linux no necesitan cambiarse pues las configuraciones requeridas para implementar IPv6 solamente deben hacerse en los archivos de configuración de los servidores; es decir, la configuración es hecha solamente a nivel de software.

Como se mencionó anteriormente, para la migración primeramente debe existir una coexistencia entre IPv4 e IPv6, razón por la cual todos los equipos, incluyendo a los host, deben soportar esta coexistencia dual.

La mayoría de usuarios usan los sistemas operativos de Microsoft que es generalmente más popular que Linux, Windows XP, Windows Server 2003 y Windows Vista soportan una arquitectura dual de IPv4 e IPv6 pero la diferencia es que en Windows Server 2003 y Windows Vista IPv6 viene instalado y listo para configurar y usarse; en cambio en Windows XP no sucede así pues se necesitan

[44] http://cisco.com/en/US/products/ps5880/index.html

instalar componentes adicionales que se los encuentra gratuitamente a través del Internet en el sitio web de Microsoft http://www.microsoft.com y también en http://www.microsoft.com/latam/windowsxp/default.mspx. Algunos componentes vienen también en el service pack 3 de Windows XP, aunque no todos.

El soporte para IPsec en Windows XP y en Windows Server 2003 es limitado, de modo que se requieren ciertas configuraciones adicionales para habilitar ciertos parámetros que sean requeridos. Hay que señalar que esto está más orientado a los hosts que utilizan Windows y con el uso de IPv6 la implementación de IPsec es obligatoria.

3.4 IPV6 EN REDES IPV4

Para implementar IPv6 en una red que funciona con IPv4 se debe tener en cuenta que la red en un principio debe soportar ambos protocolos durante el proceso que conlleva la migración. Las configuraciones que se proponen a partir de aquí son configuraciones de ejemplo para los diferentes requerimientos que se necesitan para iniciar el proceso de migración.

3.4.1 TÚNELES

El método que más se recomienda es el uso de túneles, en especial para la interconexión de redes pequeñas con redes grandes. Los tipos de túneles que generalmente se deben usar son: túneles estáticos, túneles 6to4 y túneles ISATAP. Para los routers de los enlaces troncales del ISP se podrían usar túneles 6to4, ya que este tipo de túnel interconecta dos redes IPv6 aisladas a través de IPv4. Para el router principal no existe problema con alguna configuración de túnel pues el proveedor del enlace internacional cuenta actualmente con una red funcionando con IPv4 e IPv6 de modo que en el router principal solamente bastaría configurar una dirección IPv6 y el protocolo de enrutamiento adecuado para que se establezca la comunicación.

Para los routers de los enlaces troncales se necesitaría la siguiente configuración para establecer un túnel 6to4:

Router#configure terminal
Router (config)# interface tunnel 0
Router(config-if)# ipv6 address (dirección ipv6 que se le asigne)
Router(config-if)# tunnel source (dirección ipv4 de la interfaz serial del router)
Router(config-if)# tunnel mode ipv6ip 6to4

Para transmitir todos los paquetes 6to4 a través de la interfaz 6to4 creada en el túnel, se debe añadir una ruta como se muestra a continuación:

Router#configure terminal
Router (config)#route (prefijo de la red dirección ipv6::/16) tunnel 0

En el lado del cliente Windows crea de forma automática direcciones 6to4 para todas las direcciones IPv4 asignadas a la interfaz de red con el siguiente comando:

c> netsh interface ipv6 6to4 set relay (dirección IPv4 asignada al cliente por el ISP)

Todo lo que se encuentra detrás de los routers de los enlace troncales no necesitaría configuración alguna de túneles pues se supone que toda la red interna de ISP estaría funcionando bajo IPv6.

3.4.2 ROUTING

Las direcciones de las interfaces de un router están frecuentemente relacionadas con la configuración de protocolos de enrutamiento. Desde que la autoconfiguración de una interface está basada en las direcciones MAC de la interfaz, la EUI-64 hace que las direcciones dependan del hardware de la interfaz y la sustitución de una tarjeta defectuosa genera una nueva dirección

autoconfigurada. Por lo mismo, como sugerencia la mayoría de personas expertas en IPv6 sugieren que se utilicen direcciones autoconfiguradas para las interfaces de un router sino más bien se deben configurar direcciones estáticas.

El envío de paquetes en Ipv6 no está configurado por defecto, para habilitar el envío de paquetes se debe usar el siguiente comando:

Router#configure terminal
Router (config)#ipv6 unicast-routing

En windows debe usar el siguiente commando para habilita el envoi de paquetes ipv6: ***C> netsh interface ipv6 set interface "Local Area Connection" forwarding=enabled***

En los routers Cisco se presenta la siguiente configuración para las rutas estáticas y los diferentes protocolos de enrutamiento para IPv6:

3.4.2.1 Configuración de las Rutas estáticas

Al igual que en el enrutamiento en IPv4 las rutas estáticas se configuran de manera similar en los equipos Cisco para IPv6, primero se declara que la ruta es estática, luego se declara la dirección de destino y luego la dirección o interfaz por la que saldrá el paquete:

Router#configure terminal
Router (config)#ipv6 route (dirección de destino) (dirección de salida/prefijo)

Router#configure terminal
Router (config)#ipv6 route (dirección de destino) (interfaz de salida)

La lista de todas las rutas estáticas se muestra con el comando: ***Router#show ipv6 route static.***

3.4.2.2 Configuración de Rip

El protocolo RIP para IPv6 de habilita con el siguiente comando:

Router#configure terminal
Router (config)#ipv6 router rip (nombre del proceso)
Router (config)#interface (interface en la que se habilita el protocolo)
Router (config-if)#ipv6 rip (nombre del proceso) enable

Para configurar una ruta por defecto se usa el comando:

Router#configure terminal
Router (config)#interface (interface en la que se habilita el protocolo)
Router (config-if)#ipv6 rip (nombre del proceso) default-information originate

Para incluir las rutas estáticas en los mensajes de RIP se usa el comando

Router#configure terminal
Router (config)#ipv6 router rip (nombre del proceso) redistribute static

3.4.2.3 Configuración de OSPF

El protocolo de enrutamiento OSPF para IPv6 es similar que en IPv4 y se habilita de la siguiente forma:

Router#configure terminal
Router (config)#ipv6 router ospf
Router (config)#interface ethernet 0
Router (config-if)#ipv6 ospf

Para ver la información de OSPF se usa: ***Router#show ipv6 ospf***, y para reiniciar los cálculos de “el camino más corto (SPF)” se usa el comando: ***Router#clear ipv6 ospf force-spf.***

3.4.2.4 Configuración de IS-IS

A diferencia de RIP y OSPF, IS-IS lleva las rutas IPv6 e Ipv4 en el mismo protocolo y en el mismo proceso y por esta razón muy pocos comandos se usan para ipv6:

Router#configure terminal
Router (config)#router isis (area….)
Router (config)#net (dirección de red ipv6)
Router (config)#address-family ipv6
Router (config)#interface (interface en la que se habilita el protocolo)
Router (config-if)#ipv6 router isis (area….)

Para propagar rutas por defecto en IS-IS se usa el comando:

Router#configure terminal
Router (config)#router isis (area…)
Router (config)#address-family ipv6
Router (config)#default-information originate

IS-IS comprueba de manera predetermina las adyacencias para un mismo conjunto de protocolos; cuando esta comprobación falla se debe quitar la adyacencia para permitir una muti-topología con el siguiente comando:

Router#configure terminal
Router (config)#router isis (area…)
Router (config-router)#address-family ipv6
Router (config-router-af)#no adjacency-check

Para mostrar información sobre IS-IS se usa el comando: ***Router#show isis (area…)***

3.4.2.5 Configuración de BGP

BGP se configura de la siguiente manera:

Router #configure terminal
Router (config)#router bgp (numero del sistema autónomo)
Router (config-router)#bgp router-id (dirección IPv4 del router de borde)
Router (config-router)#neighbor (dirección IPv6 del vecino) remote-as (sistema autónomo del vecino)
Router (config-router)#address-family ipv6
Router (config-router-af)#neighbor (dirección IPv6 del vecino)activate
Router (config-router-af)#network (dirección de red del vecino)

Cuando se intercambian solamente rutas IPv6 se debe usar el siguiente comando:

Router#configure terminal
Router (config)#router bgp (sistema autónomo)
Router (config-router)#no bgp default ipv4-unicast

para obtener información acerca del protocolo BGP se usa el comando: ***Router#show bgp ipv6***. Para obtener información acerca de la tabla de enrutamiento del router se usa el comando: ***Router#show ipv6 route.***

3.4.3 VECINOS IPV6

En los routers Cisco la tabla de vecinos se muestra con el comando: ***Router#show ipv6 neighbors***, y para borrar la tabla se usa el comando: ***Router#clear ipv6 neighbors.***

Para añadir un vecino se usa ***Router (config)#ipv6 neighbor*** y a continuación se añade la siguiente información: dirección ipv6 del vecino, interfaz o nombre para alcanzar al vecino, dirección de capa de Enlace del vecino.

3.4.4 ICMP

Para enviar una solicitud de eco ICMP a un nodo se usa el comando: ***Router#ping ipv6 (dirección ipv6)***. El ping extendido también está disponible.

El IOS de Cisco permite limitar el intervalo de tiempo en el que se envían los mensajes ICMP con el fin de evitar los ataque de denegación de servicio con el siguiente comando:

Router#configure terminal
Router (config)#ipv6 icmp error-interval (intervalo de tiempo en milisegundos)

3.4.5 DNS

Un router Cisco puede resolver nombres en IPv6 sin ningún problema siempre y cuando esto no afecte en el procesamiento de su función principal. Lo realiza con el siguiente comando:

Router#configure terminal
Router (config)#ip name-server (dirección ipv6 del servidor)

Se debe tener en cuenta que un nombre de host se puede poner en TFTP, SSH o telnet en donde primero se resolverá el nombre a direcciones IPv6 siempre y cuando exista un registro AAAA. El IOS de Cisco soporta la resolución de nombres a direcciones y de direcciones a nombres.

La configuración de DNS para el servidor en Linux se trata en uno de los anexos.

3.4.6 SEGURIDAD

En los equipos Cisco se puede hacer un filtrado de campos como dirección de origen y destino, protocolo de transporte del puerto de origen y destino, diffserv, ICMP, fragmentos de cabecera, etc de cada paquete IPv6. Por ejemplo para filtraR el tráfico por dirección o por algún puerto se puede utilizar una lista de acceso con el siguiente comando:

Router#configure terminal
Router (config)#ipv6 access-list (nombre de la lista de acceso)

Se permite o deniega según los requerimientos de la red. También se pueden configurar seguridades más avanzadas para IPv6 mediante el SDM (Security Device Manager) de Cisco de forma gráfica.

En Windows, el paquete de seguridades se activa con el siguiente comando:

C> netsh interface ipv6 set privacy state=enabled

Esto viene incluído en el service pack 2 de Windows XP y en los sistemas operatives Windows posteriors a XP. En Windows Server 2003 se puede realizar la configuración de todas las seguridades gráficamente una vez activado con el comando.

3.4.7 CONFIGURACIÓN DE SNMP

En el lado del proveedor el protocolo SNMP es importante pues es a través de este protocolo se obtienen las estadísticas del tráfico del ISP. SNMP en Cisco se configura de la siguiente forma:

Router#configure terminal
Router (config)#snmp-server community (nombre de la comunidad snmp)
Router (config)#snmp-server enable traps
Router (config)#snmp-server host (dirección ipv6 del servidor)

Ahora MRTG no IPv6, para poder habilitarlo se debe incluir la siguiente línea en el archivo de configuración de MRTG que se encuentra en:

/etc/mrtg/mrtg.cfg :

EnableIPv6: Yes

Luego se ingresa el siguiente comando desde el directorio raíz: ***cfgmaker –enable-ipv6***; y luego se reinicia el servicio de SNMP con el comando: ***snmpd restart*** y, por último se reinicia el servicio apache para que empiecen a llegar reportes de las estadísticas de red del ISP: ***servide httpd restart.***

3.5 SERVICIOS

El ISP brinda servicio de hosting, transferencia de archivos y correo electrónico. Estos tienen soporte para IPv6 en sus archivos de configuración ya que estos servicios están sobre Linux y este sistema operativo ofrece muchas facilidades para la implementación de IPv6 en los servicios, de modo que su actualización no resulta complicada.

3.5.1 CONFIGURACIÓN DE HOSTING Y WEB SERVER

Para el servicio de hosting se usa Apache Web Server. Para configurarlo en IPv6 se requieren de dos pasos: Primero, el servidor debe tener una dirección IPv6 y esa dirección debe estar registrada en el DNS usando registros AAAA; en segundo lugar debe estar configurado para que escuche a las direcciones IPv6. En el archivo de configuración ***httpd.conf*** del Apache web Server, la directiva ***Listen*** es utilizada para configurar la dirección y el puerto en el que escucha el servidor Web. La dirección IPv6 es delimitada por corchetes; por ejemplo para escuchar la dirección 3ffe:b00:1:1::1 la directiva ***Listen*** estaría de la siguiente forma[45]:

 # cat httpd.conf
Listen [3ffe:b00:1:1::1]

[45] BLANCHET Marc. Migrating to Ipv6, Quebec Canada, página 381.

De la misma forma, el alojamiento virtual está especificado por las directivas ***NameVirtualHost*** y ***VirtualHost*** las cuales en la configuración para especificar una dirección IPv6 también se las encierra en corchetes como se muestra a continuación:

 # cat httpd.conf
NameVirtualHost 192.0.2.1
NameVirtualHost [3ffe:b00:1:1::1]

Hay que señalar que la directiva ***NameVirtualHost*** solo puede tomar una dirección IP por línea, en cambio la directiva ***VirtualHost*** puede tomar todas las direcciones en la misma línea como se muestra a continuación[46]:

 # cat httpd.conf
<VirtualHost 192.0.2.1 [3ffe:b00:1:1::1]>
</VirtualHost>

3.5.2 CONFIGURACIÓN DEL CORREO ELECTRÓNICO

Existen varios agentes de correo, el que más se usa es el Sendmail. Este está habilitado tanto para escuchar IPv4 como IPv6 en todas las direcciones configuradas por ***Daemon_Options*** en donde se definen dos declaraciones, una para IPv6 y otra para IPv4, como se muestra a continuación:

DAEMON_OPTIONS('Name=IPv4, Family=inet')dnl
DAEMON_OPTIONS('Name=IPv6, Family=inet6')dnl

Para poder permitir que Sendmail escuche una dirección IPv6 específica se realiza la siguiente declaración en la directiva:

DAEMON_OPTIONS('Name=IPv6, Family=inet6,Addr=3ffe:b00:1:1::1')dnl

---

[46] BLANCHET Marc. Migrating to Ipv6, Quebec Canada, página 382.

En los archivos de configuración de Sendmail como ***mailertables***, ***access*** y ***relay – domains***, las direcciones IPv6 son especificadas por la palabra ***IPv6*** seguido del signo ":" y luego la dirección IPv6 como se muestra a continuación:

IPV6:3ffe:b00:1:1::1

3.5.3 OTROS SERVICIOS

Con la experiencia que muchos de los países desarrollados (principalmente en Europa y Asia) han adquirido con IPv6, se han desarrollado servicios relacionados con IPv6 que un proveedor de Internet podría prestar a sus clientes. A continuación se describen algunos de los servicios que muchos ISPs han implementado y que han resultado en beneficio tanto para el ISP como para los clientes.

El servicio de tunneling proporciona una conectividad para los usuarios de un ISP mediante túneles IPv6 sobre IPv4. Este servicio se puede ofrecer a través de una variedad de tecnologías de acceso incluyendo a las líneas dedicadas, ISDN, ADSL y por fibra óptica. Muchos proveedores han expandido el alcance de los servicios de Gateway de IPv6 a los usuarios, realizando ofertas comerciales disponibles independientes a la conexión del Internet que les brinda el proveedor. Con el uso de un router capaz de soportar tráfico IPv6 los clientes sin una conexión directa al ISP pueden recibir servicio de tunneling en IPv6[47].

El servicio de ADSL IPv6 es un servicio de doble conectividad IPv4/IPv6 a través de una línea de abonado digital asimétrica (ADSL). Los prefijos de red se asignan automáticamente al equipo CPE[48].

---

[47] AMOSS John, MINOLI Daniel. IPv4 to IPv6 Transition, New York USA, página 155.

[48] AMOSS John, MINOLI Daniel. IPv4 to IPv6 Transition, New York USA, página 156.

El servicio de multicast es un servicio que provee la multidifusión de video que el ISP podría brindar a las empresas. El servicio permite el streaming de video en tiempo real y con un mínimo de carga para el servidor de video[49].

Otro servicio que ha sido muy difundido es el de VoIP. Con el desarrollo de IPv6 se ha conseguido una mejor calidad y solidez en su transmisión en redes interoperables. Las empresas celulares están empezando a apoyar a los teléfonos basados en IP para su servicio celular. Inclusive la movilidad sobre IPv6 (MIPv6) se está haciendo popular en las redes celulares UMTS en Europa y Japón, pues valiéndose de esta el ISP podría dar también la facilidad al usuario de controlar ciertos procesos en su hogar, ya sea a través de su computador personal, palm, o su teléfono celular. En base a todo lo que implica VoIP (y más aún con el desarrollo de nuevos protocolos para VoIP en IPv6) la mayoría de proveedores están tratando de explotar al máximo este servicio ya que genera nuevas fuentes de ingresos, mismos que reemplazan de forma exitosa a los ingresos obtenidos por los servicios tradicionales.

La convergencia de redes es otro punto que el proveedor de Internet debería analizar pues aquí se integran varios servicios como la telefonía tradicional, VoIP, servios de wireless, entrega de video, etc.

En este capitulo se ha visto acerca de recomendaciones que el ISP podría seguir para poner en marcha la migración, aclarando que es un proceso que requiere de tiempo. Sin embargo, se ha podido ver que la implementación de IPv6 esta siendo cada vez más requerida debido a las prestaciones que ofrece en las comunicaciones. En el siguiente capitulo se verán las conclusiones extraídas del proyecto así como las recomendaciones que se obtuvieron.

[49] AMOSS John, MINOLI Daniel. IPv4 to IPv6 Transition, New York USA, página 156.

CAPÍTULO 4

CONCLUSIONES Y RECOMENDACIONES

CAPÍTULO 4

CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- La seguridad de todo el tráfico que pasa por el enlace internacional del ISP es administrada por la empresa Megadatos. El ISP no posee en su red interna esquemas de seguridad como; listas de acceso, firewall o NAT. El único firewall que el ISP posee lo administra Megadatos remotamente; por lo tanto, la implementación de un esquema o plan de seguridad por parte del ISP frente a una migración es un requerimiento inicial importante.

- El espacio de 128 bits que IPv6 posee para las direcciones es cuatro veces más grande que el espacio para IPv4. Con tal cantidad de direcciones en IPv6 cada habitante de la tierra tendría su propia dirección y aún así seguirían existiendo direcciones IPv6 libres. Se estima que habrían $6.65x10^{23}$ direcciones IPv6 por cada metro cuadrado en la Tierra.

- La cabecera de IPv6 tiene un nuevo formato pues se han eliminado campos no esenciales que existían en IPv4 con el fin de reducir el procesamiento de los paquetes que hacen equipos como los routers; para minimizar la sobrecarga en los procesos de red.

- La autoconfiguración de la direcciones en IPv6 es una nueva característica muy importante porque facilita el manejo de la red y la configuración por parte de los usuarios. La característica de autoconfiguración es un proceso flexible y permite generar una dirección IPv6 automáticamente a una PC local en ausencia de un servidor DHCPv6 o router.

- Características en IPv6 como la seguridad, un mejor soporte en la calidad de servicio, la eficiencia en la infraestructura de enrutamiento, etc. hacen que IPv6 sea un protocolo, llamativo y simple de configurar.

- Con todas las características que IPv6 posee en su diseño es posible continuar con el desarrollo del protocolo implementando mejoras que según el desarrollo y avance tecnológico que puedan requerirse a futuro, y así cumpla con los requerimientos crecientes de las comunicaciones.

- La migración a IPv6 es posible y factible pero es un proceso que lleva tiempo. Del trabajo aquí realizado se puede afirmar que los cambios que se requiere son a nivel de software y la mayoría de equipos en la actualidad cuenta con soporte para IPv6.

- Para lograr la migración se debe seguir un proceso en el cual el primer paso inevitable es la coexistencia con IPv4.

- Los cambios que actualmente se están dando en el desarrollo de la comunicaciones con la creación de dispositivos portátiles, de entretenimiento, y equipos tanto para el hogar como para la empresa hacen que IPv6 sea cada vez más requerido debido a los servicios que se pueden obtener y de allí la rentabilidad.

4.2 RECOMENDACIONES

- Antes de realizar la migración se debe considerar aspectos importantes que permitan diseñar un plan, teniendo en cuenta parámetros como el tamaño de la red, la asignación de direcciones IPv6, la actualización de los equipos y sobre todo la seguridad y las herramientas que van a ser

utilizadas para monitorear la red una vez que la migración se ponga en marcha.

- Existen varias recomendaciones de entidades internacionales que han sido elaboradas específicamente para los ISPs. Si bien estas guías no necesariamente deben ser cumplidas al pie de la letra, si constituyen una buena guía de lo que se debe tener en cuenta para la migración.

- Los routers de los enlaces troncales del ISP deben necesariamente ser reemplazados para la migración pues no permiten la actualización del IOS.

- El router Cisco 2821 es el que puede reemplazar a los routers de los enlaces troncales y si el proveedor quiere empezar a brindar servicios de VoIP con IPv6, este equipo tiene el soporte adecuado para esta funcionalidad.

- La mayoría de ISPs en Europa, Asia y Norte América han empezado a migrar sus redes a IPv6; fruto de esta migración han lanzado servicios sobre IPv6 que han resultado muy convenientes y por esta razón los servicios descritos en el Capítulo 3 se presentan como una recomendación que los demás proveedores, de acuerdo a la demanda tecnológica del país, podrían adoptar.

- La migración se compone de muchos pasos entre los cuales primero se debe diseñar un esquema que permita que IPv6 conviva con IPv4; esto quiere decir que los equipos, como los routers, deben estar en la capacidad de trabajar con los dos protocolos.

REFERENCIAS BIBLIOGRÁFICAS

1. Braun Marcelo, Bragnulo, Herramientas para la conectividad IPv6 con múltiples proveedores. Universidad Carlos III. Leganés España. Mayo 2005.

2. Castro Eva, Interoperabilidad de Aplicaciones IPv4 e IPv6. Universidad Rey Juan Carlos. Madrid España. 2004.

3. Amoss Jhon, Minoli Daniel, Handbook of IPv4 to IPv6 Transition. Primera Edición. Auerbach Publications. New York USA. 2008.

4. Qing Li, Jinmey Tatuya, Shima Keiichi, Ipv6 Advanced Protocols Implementation. Primera Edición. San Francisco California, USA. 2007.

5. Blanchet Marc, Migrating to IPv6. Segunda Edición. John Wiley & Sons Ltd. Quebec Canada. 2007.

6. Davies Joseph, Understanding IPv6. Primera Edición. Microsoft Press. Washington USA. 2003.

7. Oleas Juan, Estudio y análisis para la migración de IPv4 a IPv6. Escuela Politécnica Nacional. Quito Ecuador. 2001.

8. http://www.cu.ipv6tf.org/pdf/ipv6-UNLP.PDF

9. http://www.it.uc3m.es/netcom/docs/thesis_marcelo.pdf

10. http://internetng.dit.upm.es/papers/InternetNGv10.pdf

ANEXOS

ANEXO A

SUBNETTING THE IPV6 ADDRESS SPACE

Just as in IPv4, the IPv6 address space can be divided by using high-order bits that do not already have fixed values to create subnetted network prefixes. These are used either to summarize a level in the routing or addressing hierarchy (with a prefix length less than 64), or to define a specific subnet or network segment (with a prefix length of 64). IPv4 subnetting differs from IPv6 subnetting in the definition of the host ID portion of the address. In IPv4, the host ID can be of varying length, depending on the subnetting scheme. For currently defined unicast IPv6 addresses, the host ID is the interface ID portion of the IPv6 unicast address and is always a fixed size of 64 bits.

SUBNETTING FOR NLA IDS

If you are an ISP, subnetting the IPv6 address space consists of using subnetting techniques to divide the NLA ID portion of a global address in a manner that allows for route summarization and delegation of the remaining address space for different portions of your network, for downstream providers, or for individual customers. The global address has a 24-bit NLA ID field to be used by the various layers of ISPs between a top-level aggregator (a global ISP identified by the TLA ID) and a customer site.

For a global address allocated to a top-level aggregator, the first 16 bits of the address are fixed and correspond to the FP (set to 001) and the TLA ID (13 bits in length). The TLA ID is followed by the Res portion, which consists of 8 reserved bits set to 0. Therefore, for subnetting of the NLA ID portion of a global address, the first 24 bits are fixed. In a global address, the Res bits are never shown due to the suppression of leading zeros in IPv6 colon hexadecimal notation.

Subnetting the NLA ID portion of a global address requires a two-step procedure:

1. Determine the number of bits to be used for the subnetting.
2. Enumerate the new subnetted network prefixes.

The subnetting technique described here assumes that subnetting is done by dividing the 24-bit address space of the NLA ID using the high-order bits in the NLA ID that do not already have fixed values. While this method promotes hierarchical addressing and routing, it is not required. For example, you can also create a flat addressing space for the NLA ID by numbering the subnets from 0 to 16,777,215.

STEP 1: DETERMINING THE NUMBER OF SUBNETTING BITS

The number of bits being used for subnetting determines the possible number of new subnetted network prefixes that can be allocated to portions of your network based on geographical, customer segment, or other divisions. In a hierarchical routing infrastructure,

you need to determine how many network prefixes, and therefore how many bits, you need at each level in the hierarchy. The more bits you choose for the various levels of the hierarchy, the fewer bits you will have available to enumerate individual subnets in the last level of the hierarchy. The last level in the hierarchy is used to assign 48-bit prefixes to customer sites.

For example, a network designer at a large ISP decides to implement a two-level hierarchy reflecting a geographical/customer segment structure and uses 8 bits for the geographical level and 8 bits for the customer segment level. This means that each customer segment in each geographical location has only 8 bits of subnetting space left (24 - 8 - 8), or only 256 (= 28) 48-bit prefixes per customer segment.

On any given level in the hierarchy, you will have a number of bits that are already fixed by the next level up in the hierarchy (*f*), a number of bits used for subnetting at the current level in the hierarchy (*s*), and a number of bits remaining for the next level down in the hierarchy (*r*). At all times, $f + s + r = 24$. This relationship is shown in Figure 1.

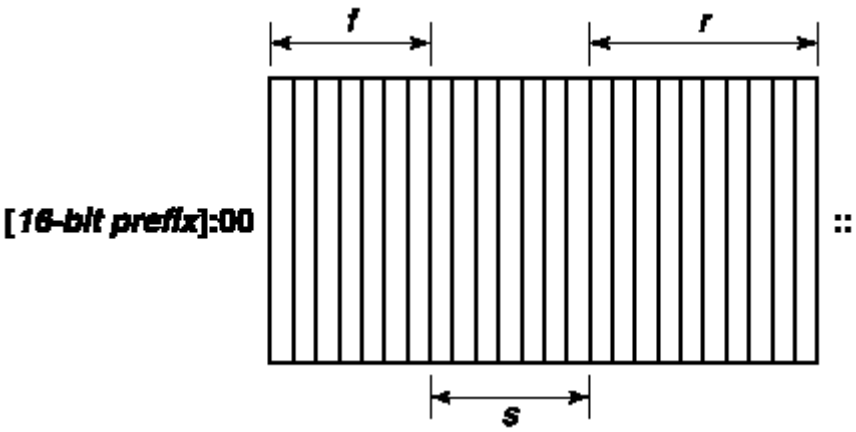

Figure 1. *The subnetting of an NLA ID*

STEP 2: ENUMERATING SUBNETTED NETWORK PREFIXES

Based on the number of bits used for subnetting, you must list the new subnetted network prefixes. There are two main approaches:

- Hexadecimal — Enumerate new subnetted network prefixes by using hexadecimal representations of the NLA ID and increment.
- Decimal — Enumerate new subnetted network prefixes by using decimal representations of the NLA ID and increment. The decimal subnetting technique is included here for those who are more comfortable dealing with decimal numbers ($Base_{10}$).

Either method produces the same result: an enumerated list of subnetted network prefixes.

Creating the enumerated list of subnetted network prefixes by using the hexadecimal method

1. Based on *s* (the number of bits chosen for subnetting), and *m* (the prefix length of the network prefix being subnetted), calculate the following:

   $f = m - 24$

   f is the number of bits within the NLA ID that are already fixed.

   $n = 2^s$

   n is the number of network prefixes that are obtained.

   $i = 2^{24-(f+s)}$

   i is the incremental value between each successive NLA ID expressed in hexadecimal form.

   $l = 24 + f + s$

   l is the prefix length of the new subnetted network prefixes.

2. Create a three-column table with *n* entries. The first column is the network prefix number (starting with 1), the second column is the value of *F* (the hexadecimal representation of the NLA ID), and the third column is the new subnetted network prefix.
3. In the first table entry, the entry for the NLA ID column is *F* and the subnetted network prefix is the original network prefix with the new prefix length. To obtain *F*, combine the last two hexadecimal digits of the second hexadecimal block with the four hexadecimal digits of the third hexadecimal block of the NLA ID being subnetted to form a 6-digit hexadecimal number. Remember to include zeros that may not be present due to leading zero suppression. For example, for the global address prefix 3000:4D:C00::/38, *F* is 0x4D0C00.
4. In the next table entry, for the NLA ID column, increase the value of *F* by *i*. For example, in the second table entry, the NLA ID is *F* + *i*.
5. For the subnetted network prefix column, convert the NLA ID into two separate 16-bit blocks in colon hexadecimal notation and place them after the 16-bit prefix to express the new subnetted network prefix. For example, for the second table entry, the subnetted network prefix is [*16-bit prefix*]:[*F* + *i* (expressed in colon hexadecimal notation)]::/*l*.
6. Repeat steps 4 and 5 until the table is complete.

For example, to perform a 3-bit subnetting of the global network prefix 3000:4D:C00::/38, we first calculate the values of the number of prefixes, the increment, and the new prefix length. Our starting values are *F* = 0x4D0C00, *s* = 3, and *f* = 38 - 24 = 14. The number of

prefixes is 8 ($n = 2^3$). The increment is 0x80 ($i = 2^{24-(14+3)} = 128 = 0x80$). The new prefix length is 41 ($l = 38 + 3$).

Next, we construct a table with 8 entries. The subnetted network prefix for network prefix 1 is 3000:4D:C00::/41. Additional entries in the table are successive increments of *i* in the NLA ID portion of the network prefix, as shown in Table 1.

Table 1. The Hexadecimal Subnetting Technique for Network Prefix 3000:4D:C00::/38

Network Prefix Number	***NLA ID (hexadecimal)***	***Subnetted Network Prefix***
1	4D0C00	3000:4D:C00::/41
2	4D0C80	3000:4D:C80::/41
3	4D0D00	3000:4D:D00::/41
4	4D0D80	3000:4D:D80::/41
5	4D0E00	3000:4D:E00::/41
6	4D0E80	3000:4D:E80::/41
7	4D0F00	3000:4D:F00::/41
8	4D0F80	3000:4D:F80::/41

NOTE

RFC 2373 allows the use of subnetted network prefixes where the bits being used for subnetting are set to all zeros (the all-zeros subnetted network prefix) and all ones (the all-ones subnetted network prefix) for any portion of the IPv6 network prefix being subnetted.

Creating the enumerated list of subnetted network prefixes using the decimal method

1. Based on *s* (the number of bits chosen for subnetting), and *m* (the prefix length of the network prefix being subnetted), and *F* (the hexadecimal value of the NLA ID being subnetted), calculate the following:

   $f = m - 24$

   f is the number of bits within the NLA ID that are already fixed.

   $n = 2^s$

   n is the number of network prefixes that are obtained.

$i = 2^{24-(f+s)}$

i is the incremental value between each successive NLA ID expressed in decimal form.

$l = 24 + f + s$

l is the prefix length of the new subnetted network prefixes.

D = decimal representation of *F*

2. Create a four-column table with *n* entries. The first column is the network prefix number (starting with 1), the second column is the decimal representation of the NLA ID portion of the new subnetted network prefix, the third column is the hexadecimal representation of the NLA ID portion of the new subnetted network prefix, and the fourth column is the new subnetted network prefix.
3. In the first table entry, the decimal representation of the NLA ID is *D*, the hexadecimal representation of the NLA ID is *F*, and the subnetted network prefix is the original network prefix with the new prefix length.
4. In the next table entry, for the second column, increase the value of the decimal representation of the NLA ID by *i*. For example, in the second table entry, the decimal representation of the subnet ID is *D* + *i*.
5. For the third column, convert the decimal representation of the NLA ID to hexadecimal.
6. For the fourth column, convert the NLA ID into two separate 16-bit blocks in colon hexadecimal notation and place them after the 16-bit prefix to express the new subnetted network prefix. For example, for the second table entry, the subnetted network prefix is [*16-bit prefix*]:[*F* + *i* (expressed in colon hexadecimal notation)]::/*l*.
7. Repeat steps 4 through 6 until the table is complete.

For example, to perform a 3-bit subnetting of the global network prefix 3000:4D:C00::/38, we first calculate the values of the number of prefixes, the increment, and the new prefix length. Our starting values are *F* = 0x4D0C00, *s* = 3, and *f* = 38 - 24 = 14. The number of prefixes is 8 ($n = 2^3$). The increment is 128 ($i = 2^{24-(14+3)} = 128$). The new prefix length is 41 (*l* = 38 + 3). The decimal representation of the starting NLA ID is 5049344 (*D* = 0x4D0C00 = 5049344).

Next, we construct a table with 8 entries. The subnetted network prefix for network prefix 1 is 3000:4D:C00::/41. Additional entries in the table are successive increments of *i* in the NLA ID portion of the network prefix, as shown in Table 2.

Table 2. *The Decimal Subnetting Technique for Network Prefix 3000:4D:C00::/38*

Network Prefix Number	***Decimal Representation of NLA ID***	***Hexadecimal Representation of NLA ID***	***Subnetted Network Prefix***
1	5049344	4D0C00	3000:4D:C00::/41
2	5049472	4D0C80	3000:4D:C80::/41
3	5049600	4D0D00	3000:4D:D00::/41
4	5049728	4D0D80	3000:4D:D80::/41
5	5049856	4D0E00	3000:4D:E00::/41
6	5049984	4D0E80	3000:4D:E80::/41
7	5050112	4D0F00	3000:4D:F00::/41
8	5050240	4D0F80	3000:4D:F80::/41

SUBNETTING FOR SLA IDS/SUBNET IDS

For most network administrators within an organization, subnetting the IPv6 address space consists of using subnetting techniques to divide the SLA ID portion of the global address or the Subnet ID portion of the site-local address in a manner that allows for route summarization and delegation of the remaining address space to different portions of an IPv6 intranet. The global address has a 16-bit SLA ID field to be used by organizations within their sites. The site-local address has a 16-bit Subnet ID field to be used by organizations within a site.

In both cases, the first 48 bits of the address are fixed. For the global address, the first 48 bits are fixed and allocated by an ISP and correspond to the TLA and NLA ID portions of the global address. For the site-local address, the first 48 bits are fixed at FEC0::/48. In the discussion that follows, the term subnet ID refers to either the SLA ID portion of the global address or the Subnet ID portion of a site-local address.

Subnetting the subnet ID portion of a global or site-local address space requires a two-step procedure:

1. Determine the number of bits to be used for the subnetting.

2. Enumerate the new subnetted network prefixes.

The subnetting technique described here assumes that subnetting is done by dividing the 16-bit address space of the subnet ID using the high-order bits in the subnet ID. While this method promotes hierarchical addressing and routing, it is not required. For example, in a

small organization with a small number of subnets, you can also create a flat addressing space for the subnet ID by numbering the subnets starting at 0.

As described in the "Local-Use Unicast Addresses" section of this chapter, you can use the same subnetting scheme and use the same subnet ID for both site-local and global address network prefixes.

STEP 1: DETERMINING THE NUMBER OF SUBNETTING BITS

The number of bits being used for subnetting determines the possible number of new subnetted network prefixes that can be allocated to portions of your network based on geographical or departmental divisions. In a hierarchical routing infrastructure, you need to determine how many network prefixes, and therefore how many bits, you need at each level in the hierarchy. The more bits you choose for the various levels of the hierarchy, the fewer bits you will have available to enumerate individual subnets in the last level of the hierarchy.

For example, a network administrator decides to implement a two-level hierarchy reflecting a geographical/departmental structure and uses 4 bits for the geographical level and 6 bits for the departmental level. This means that each department in each geographical location has only 6 bits of subnetting space left (16 - 6 - 4), or only 64 (= 2^6) subnets per department.

On any given level in the hierarchy, you will have a number of bits that are already fixed by the next level up in the hierarchy (*f*), a number of bits used for subnetting at the current level in the hierarchy (*s*), and a number of bits remaining for the next level down in the hierarchy (*r*). At all times, $f + s + r = 16$. This relationship is shown in Figure 2.

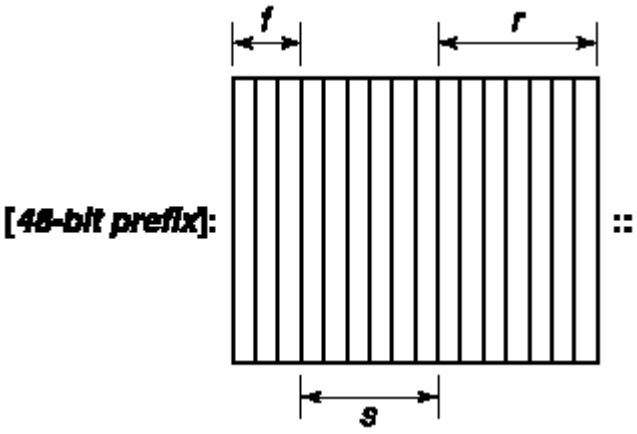

Figure 2. *The subnetting of a Subnet ID*

STEP 2: ENUMERATING SUBNETTED NETWORK PREFIXES

Based on the number of bits used for subnetting, you must list the new subnetted network prefixes. There are two main approaches:

- Hexadecimal — Enumerate new subnetted network prefixes by using hexadecimal representations of the subnet ID and increment.
- Decimal — Enumerate new subnetted network prefixes by using decimal representations of the subnet ID and increment.

Either method produces the same result: an enumerated list of subnetted network prefixes.

Creating the enumerated list of subnetted network prefixes using the hexadecimal method

1. Based on *s* (the number of bits chosen for subnetting), *m* (the prefix length of the network prefix being subnetted), and *F* (the hexadecimal value of the subnet being subnetted), calculate the following:

   $f = m - 48$

   f is the number of bits within the subnet ID that are already fixed.

   $n = 2^s$

   n is the number of network prefixes that are obtained.

   $i = 2^{16-(f+s)}$

   i is the incremental value between each successive subnet ID expressed in hexadecimal form.

   $l = 48 + f + s$

   l is the prefix length of the new subnetted network prefixes.

2. Create a two-column table with *n* entries. The first column is the network prefix number (starting with 1) and the second column is the new subnetted network prefix.
3. In the first table entry, based on *F*, the hexadecimal value of the subnet ID being subnetted, the subnetted network prefix is [*48-bit prefix*]:*F*::/*l*.
4. In the next table entry, increase the value within the subnet ID portion of the site-local or global address by *i*. For example, in the second table entry, the subnetted prefix is [*48-bit prefix*]:*F* + *i*::/*l*.
5. Repeat step 4 until the table is complete.

For example, to perform a 3-bit subnetting of the site-local network prefix FEC0:0:0:C000::/51, we first calculate the values of the number of prefixes, the increment, and the new prefix length. Our starting values are F = 0xC000, s = 3, and f = 51 - 48 = 3. The number of prefixes is 8 ($n = 2^3$). The increment is 0x400 ($i = 2^{16-(3+3)} = 1024$ = 0x400). The new prefix length is 54 (l = 48 + 3 + 3).

Next, we construct a table with 8 entries. The entry for the network prefix 1 is FEC0:0:0:C000::/54. Additional entries in the table are successive increments of i in the subnet ID portion of the network prefix, as shown in Table 3.

Table 3. *The Hexadecimal Subnetting Technique for Network Prefix FEC0:0:0:C000::/51*

Network Prefix Number	***Subnetted Network Prefix***
1	FEC0:0:0:C000::/54
2	FEC0:0:0:C400::/54
3	FEC0:0:0:C800::/54
4	FEC0:0:0:CC00::/54
5	FEC0:0:0:D000::/54
6	FEC0:0:0:D400::/54
7	FEC0:0:0:D800::/54
8	FEC0:0:0:DC00::/54

Creating the enumerated list of subnetted network prefixes using the decimal method

1. Based on *s* (the number of bits chosen for subnetting), and *m* (the prefix length of the network prefix being subnetted), and *F* (the hexadecimal value of the subnet ID being subnetted), calculate the following:

   $f = m - 48$

   f is the number of bits within the subnet ID that are already fixed.

   $n = 2^s$

   n is the number of network prefixes that are obtained.

   $i = 2^{16-(f+s)}$

   i is the incremental value between each successive subnet ID.

   $l = 48 + f + s$

l is the prefix length of the new subnetted network prefixes.

D = decimal representation of *F*

2. Create a three-column table with *n* entries. The first column is the network prefix number (starting with 1), the second column is the decimal representation of the subnet ID portion of the new network prefix, and the third column is the new subnetted network prefix.
3. In the first table entry, the decimal representation of the subnet ID is *D* and the subnetted network prefix is [*48-bit prefix*]:*F*::/*l*.
4. In the next table entry, for the second column, increase the value of the decimal representation of the subnet ID by *i*. For example, in the second table entry, the decimal representation of the subnet ID is *D* + *i*.
5. For the third column, convert the decimal representation of the subnet ID to hexadecimal and construct the prefix from [*48-bit prefix*]:[*subnet ID*]::/*l*. For example, in the second table entry, the subnetted network prefix is [*48-bit prefix*]:[*D* + *i* (converted to hexadecimal)]::/*l*.
6. Repeat steps 4 and 5 until the table is complete.

For example, to perform a 3-bit subnetting of the site-local network prefix FEC0:0:0:C000::/51, we first calculate the values of the number of prefixes, the increment, the new prefix length, and the decimal representation of the starting subnet ID. Our starting values are *F* = 0xC000, *s* = 3, and *f* = 51 - 48 = 3. The number of prefixes is 8 ($n = 2^3$). The increment is 1024 ($i = 2^{16-(3+3)}$). The new prefix length is 54 (*l* = 48 + 3 + 3). The decimal representation of the starting subnet ID is 49152 (*D* = 0xC000 = 49152).

Next, we construct a table with 8 entries. The entry for the network prefix 1 is 49152 and FEC0:0:0:C000::/54. Additional entries in the table are successive increments of *i* in the subnet ID portion of the network prefix, as shown in Table 4.

Table 4. The Decimal Subnetting Technique for Network Prefix FEC0:0:0:C000::/51

Network Prefix Number	***Decimal Representation of Subnet ID***	***Subnetted Network Prefix***
1	49152	FEC0:0:0:C000::/54
2	50176	FEC0:0:0:C400::/54
3	51200	FEC0:0:0:C800::/54
4	52224	FEC0:0:0:CC00::/54
5	53248	FEC0:0:0:D000::/54
6	54272	FEC0:0:0:D400::/54
7	55296	FEC0:0:0:D800::/54

Network Prefix Number	***Decimal Representation of Subnet ID***	***Subnetted Network Prefix***
8	56320	FEC0:0:0:DC00::/54

ANEXO B

RECOMENDACIONES IPV6

Este anexo contiene las recomendaciones más relevantes de IPv6

RECOMENDACIONES GENERALES

RFC #	***Categoría***	***Título***
1752	Standards Track	The Recommendation for the IP Next Generation Protocol
1924	Informational	A Compact Representation of IPv6 Addresses
2851	Standards Track	Textual Conventions for Internet Network Addresses
	Internet draft	The Case for IPv6

DIRECCIONAMIENTO

RFC #	***Categoría***	***Título***
1881	Informational	IPv6 Address Allocation Management
1887	Informational	An Architecture for IPv6 Unicast Address Allocation
1888	Experimental	OSI NSAPs and IPv6
2373	Standards Track	IP Version 6 Addressing Architecture
	Internet draft	IP Version 6 Addressing Architecture
2374	Standards Track	An IPv6 Aggregatable Global Unicast Address Format
2375	Informational	IPv6 Multicast Address Assignments
2450	Informational	Proposed TLA and NLA Assignment Rules
2471	Experimental	IPv6 Testing Address Allocation
2526	Standards Track	Reserved IPv6 Subnet Anycast Addresses
2921	Informational	6BONE pTLA and pNLA Formats (pTLA)
2928	Informational	Initial IPv6 Sub-TLA ID Assignments
3041	Standards Track	Privacy Extensions for Stateless Address Autoconfiguration in IPv6
	Internet draft	Site prefixes in Neighbor Discovery
	Internet draft	Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
	Internet draft	A flexible method for managing the assignment of bits of an IPv6 address block

APLICACIONES

RFC #	***Categoría***	***Título***
1886	Standards Track	DNS Extensions to support IP version 6
2428	Standards Track	FTP Extensions for IPv6 and NATs
2732	Standards Track	Format for Literal IPv6 Addresses in URL's
2874	Standards Track	DNS Extensions to Support IPv6 Address Aggregation and Renumbering
	Internet draft	IPv6 Node Information Queries

SOCKETS API

RFC #	***Categoría***	***Título***
2292	Informational	Advanced Sockets API for IPv6
	Internet draft	Advanced Sockets API for IPv6
2553	Informational	Basic Socket Interface Extensions for IPv6
	Internet draft	An Extension of Format for IPv6 Scoped Addresses

CAPA DE TRANSPORTE

RFC #	***Categoría***	***Título***
2452	Standards Track	IP Version 6 Management Information Base for the Transmission Control Protocol
2454	Standards Track	IP Version 6 Management Information Base for the User Datagram Protocol
	Internet draft	The UDP Lite Protocol

CAPA DE RED

RFC #	***Categoría***	***Título***
1809	Informational	Using the Flow Label Field in IPv6
2460	Standards Track	Internet Protocol, Version 6 (IPv6) Specification
2461	Standards Track	Neighbor Discovery for IP Version 6 (IPv6)
2462	Standards Track	IPv6 Stateless Address Autoconfiguration
2463	Standards Track	Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification
	Internet draft	Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)
2465	Standards Track	Management Information Base for IP Version 6: Textual Conventions and General Group
2466	Standards Track	Management Information Base for IP Version 6: ICMPv6 Group
2474	Standards Track	Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers
2675	Standards Track	IPv6 Jumbograms
2710	Standards Track	Multicast Listener Discovery (MLD) for IPv6
2711	Standards Track	IPv6 Router Alert Option
2767	Informational	Dual Stack Hosts using the "Bump-In-the-Stack" Technique (BIS)
3019	Standards Track	IP Version 6 Management Information Base for the Multicast Listener Discovery Protocol
3122	Standards Track	Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification
	Internet draft	Mobility Support in IPv6
	Internet draft	Default Address Selection for IPv6

SEGURIDAD

RFC #	*Categoría*	*Título*
1828	Standards Track	IP Authentication using Keyed MD5
1829	Standards Track	The ESP DES-CBC Transform
2401	Standards Track	Security Architecture for the Internet Protocol
2402	Standards Track	IP Authentication Header
2403	Standards Track	The Use of HMAC-MD5-96 within ESP and AH
2404	Standards Track	The Use of HMAC-SHA-1-96 within ESP and AH
2406	Standards Track	IP Encapsulating Security Payload (ESP)

CAPA DE ENLACE

RFC #	*Categoría*	*Título*
2464	Standards Track	Transmission of IPv6 Packets over Ethernet Networks
2467	Standards Track	Transmission of IPv6 Packets over FDDI Networks
2470	Standards Track	Transmission of IPv6 Packets over Token Ring Networks
2472	Standards Track	IP Version 6 over PPP
2473	Standards Track	Generic Packet Tunneling in IPv6 Specification
2491	Standards Track	IPv6 over Non-Broadcast Multiple Access (NBMA) networks
2492	Standards Track	IPv6 over ATM Networks
2497	Standards Track	Transmission of IPv6 Packets over ARCnet Networks
2507	Standards Track	IP Header Compression
2508	Standards Track	Compressing IP/UDP/RTP Headers for Low-Speed Serial Links
2509	Standards Track	IP Header Compression over PPP
2590	Standards Track	Transmission of IPv6 Packets over Frame Relay Networks Specification

RFC #	***Categoría***	***Título***
3146	Standards Track	Transmission of IPv6 Packets over IEEE 1394 Networks

ROUTING

RFC #	***Categoría***	***Título***
2080	Standards Track	RIPng for IPv6
2185	Informational	Routing Aspects of IPv6 Transition
2283	Standards Track	Multiprotocol Extensions for BGP-4
2545	Standards Track	Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing
2740	Standards Track	OSPF for IPv6
2772	Informational	6Bone Backbone Routing Guidelines
2894	Standards Track	Router Renumbering for IPv6

COEXISTENCIA Y MIGRACIÓN

RFC #	***Categoría***	***Título***
2529	Standards Track	Transmission of IPv6 over IPv4 Domains without Explicit Tunnels
2893	Standards Track	Transition Mechanisms for IPv6 Hosts and Routers
3053	Informational	IPv6 Tunnel Broker
3056	Standards Track	Connection of IPv6 Domains via IPv4 Clouds
	Internet draft	Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)

ANEXO C

WINDOWS SOCKETS CHANGES FOR IPV6

This includes information on changes that have been made to the Windows Sockets API to support IPv6 applications. The following topics are discussed:

- Added constants
- Address data structures
- Wildcard addresses
- Core sockets functions
- Name-to-address translation
- Address-to-name translation
- Address conversion functions
- Socket options
- New macros
- Unsupported APIs

Examples of how and when to utilize these changes in an application are discussed. Additional details can be found on the Microsoft Developer Network web site at http://msdn.microsoft.com.

ADDED CONSTANTS

A new address family name for IPv6 is required so that the address structure can be correctly identified and parsed. Similarly, a new protocol family name (with the same value as the address family name) must be defined so that a socket is created using the appropriate protocol. The address family name and protocol family name constants for IPv6 are:

- AF_INET6
- PF_INET6

ADDRESS DATA STRUCTURES

The term *sockets* defines a protocol-specific data structure that holds elements of a socket address. For IPv4, this structure is *sockaddr_in*. Sockets also defines a protocol-independent structure (*sockaddr*) for the protocol-specific structures to be cast into. The identifying field (the address family) in the protocol-specific structure overlays the family field in the generic structure. Because IPv6 addresses are different than IPv4 addresses, a new protocol-specific structure for IPv6 is required.

The data structures *sockaddr* and *sockaddr_in* are the same size which could lead one into making incorrect assumptions about the size of their related address structures. The IPv6 address structure, *sockaddr_in6*, is larger by necessity. For example, the *sockaddr* structure cannot be used to allocate storage for *sockaddr_in6*. This is discussed in more detail below.

IN6_ADDR

```
struct in6_addr {

   union {

      u_char Byte[16];

      ushort Word[8];

      } u;

};
```

The socket address structure contains information above and beyond the address for the socket. One portion of the structure, however, must be the address. In IPv4's address structure, this address is contained in *in_addr*. A larger structure, *in6_addr*, has been defined to hold the larger IPv6 address.

SOCKADDR_IN6

In addition to the larger address size, there are other members that must be represented in the socket address structure for IPv6. Although the IPv4 *sockaddr_in* structure has unused space, it is not enough to contain this additional information. The *sockaddr_in6* structure is used to contain an IPv6 address.

```
struct sockaddr_in6 {

   sa_family_t     sin6_family;

   in_port_t       sin6_port;

   uint32_t        sin6_flowinfo;

   struct in6_addr sin6_addr;

   uint32_t        sin6_scope_id;

};
```

In addition to the family, port, and address information, this structure contains sin6_flowinfo and sin6_scope_id members. *sin6_flowinfo* is intended to contain the traffic class and flow label from and for the IPv6 header. *sin6_flowinfo* is not supported in Windows XP and the Windows .NET Server 2003 family. *sin6_scope_id* contains the scope ID, which is used to identify a set of interfaces that are appropriate for the address carried in the address field.

SOCKADDR_STORAGE

As mentioned earlier, *sockaddr* and *sockaddr_in6* have different sizes. Because of this, *struct sockaddr* cannot be used to allocate storage and then be cast to a *sockaddr_in6* pointer. If static allocation of storage for *sockaddr_in6* (or even *sockaddr_in*) structures is needed, *struct sockaddr_storage* should be used. Here is an example:

```
struct sockaddr_storage newaddr;

...

msgsock = accept(listen_socket,(struct sockaddr*)&newaddr, &newaddrlen);
```

In addition to being large enough to accommodate all known protocol-specific socket address structures (including *sockaddr_in6*), *sockaddr_storage* is aligned at an appropriate boundary so that protocol-specific socket address data-structure pointers can be cast to it, enabling it to access fields without experiencing alignment problems.

WILDCARD ADDRESSES

To allow the protocol implementation to choose the source address for a connection or datagram with IPv4, a constant of INADDR_ANY (the wildcard address) is used as the address in the bind() call.

The IPv6 address type (*in6_addr*) is a structure. A constant cannot be used in an assignment for this variable, but can be used to initialize the structure. Thus, we end up with two possible ways to provide the wildcard address.

The global variable, in6addr_any, can be used in an assignment. For example:

```
sin6.sin6_addr = in6addr_any;
```

Or the constant, IN6ADDR_ANY_INIT, can be used to initialize the address structure (at declaration time only). For example:

```
struct in6_addr anyaddr = IN6ADDR_ANY_INIT;
```

IN6ADDR_LOOPBACK AND IN6ADDR_LOOPBACK_INIT

Similarly, the INADDR_LOOPBACK constant is used in IPv4 connect(), send(), and sendmsg() calls to communicate with services that reside on the local node. For IPv6 loopback, a global variable (*in6addr_loopback*) is used for assignment and a constant (IN6ADDR_LOOPBACK_INIT) is used for initialization at declaration time.

NOTE

The IPv4 INADDR_*XXX* constants were defined in host-byte order. The IPv6 equivalents are defined in network-byte order.

CORE SOCKETS FUNCTIONS

An address is passed in core Sockets functions as an opaque address pointer and length. Because of this, changes need not be made to these core Sockets functions for IPv6. The application developer needs simply to supply the appropriate IPv6 address structure and family constants.

Sockets functions that pass addresses are the following:

- bind()
- connect()
- sendmsg()
- sendto()

Sockets functions that return addresses are the following:

- accept()
- recvfrom()
- recvmsg()
- getpeername()
- getsockname()

NAME-TO-ADDRESS TRANSLATION

To resolve a host name to one or more IP addresses in IPv4, the application might use gethostbyname(). This API does not allow the caller to specify anything about the types of addresses wanted and the structure contains only enough space to store an IPv4 address. To address these issues, a new API named getaddrinfo() is introduced with IPv6. This API is protocol-independent and can be used for both IPv4 and IPv6 name-to-address resolutions. The return from this call is in the form of *addrinfo* structures that can subsequently be used to both open and use a socket.

The function prototype for getaddrinfo() is the following:

int getaddrinfo(

IN const char FAR *nodename,

IN const char FAR *servname,

```
    IN const struct addrinfo FAR *hints,
    OUT struct addrinfo FAR *FAR *res
    );
struct addrinfo {
    int ai_flags;
    int ai_family;
    int ai_socktype;
    int ai_protocol;
    size_t ai_addrlen;
    char *ai_canonname;
    struct sockaddr *ai_addr;
struct addrinfo *ai_next;
};
```

As arguments, either a node name or service name (or both) are provided. The node name can (optionally) be a numeric address string and the service name can (optionally) be a decimal port number. An *addrinfo* structure can be provided (optionally) to provide hints for the type of socket that the caller supports. The *addrinfo* structure pointed to by this hints parameter can specify a preferred socket type, family and protocol, and the following flags:

- AI_PASSIVE

  This flag indicates that the caller plans to use the returned address structure in a bind() call when set, or a connect() call when not set. Setting the node name to NULL has additional meaning depending on this flag. If the node name in the hints is NULL, and this flag is set, the returned addresses will be wildcard addresses. If the node name in the hints is NULL, and this flag is not set, the returned addresses will be loopback addresses.

- AI_CANONNAME

  The AI_CANONNAME flag indicates (when set) that the first *addrinfo* structure returned should contain a null-terminated string that contains the canonical name of the node name in the ai_canonname member.

- AI_NUMERICHOST

  This flag indicates that the nodename in the call is a numeric address string.

- AI_V4MAPPED

  If the address family specified is AF_INET6, the caller will accept IPv4-mapped IPv6 addresses. The IPv6 protocol for Windows XP and the Windows .NET Server 2003 family does not support the use of IPv4-mapped addresses.

- AI_ALL

  Used with the AI_V4MAPPED flag to indicate that the caller would like all addresses, both true IPv6 addresses and IPv4-mapped IPv6 addresses. The address family specified must be AF_INET6. The IPv6 protocol for Windows XP and the Windows .NET Server 2003 family does not support the use of this flag.

- AI_ADDRCONFIG

  The AI_ADDRCONFIG flag controls whether the query requests AAAA DNS records or A records, based on the locally configured source addresses. AAAA records will be queried only if the node has at least one IPv6 source address. A records will be queried only if the node has at least one IPv4 source address.

A pointer to a linked list of *addrinfo* structures is returned. The order of the addresses is in decreasing order of desirability.

The *addrinfo* structures (and structures contained as members within those structures) are dynamically allocated and must be released by calling freeaddrinfo() with a pointer to the linked list of *addrinfo* structures.

The function prototype for freeaddrinfo() is the following:

```
void freeaddrinfo(

  struct addrinfo FAR *ai

);
```

ADDRESS-TO-NAME TRANSLATION

A reverse lookup can be performed by using another new Sockets function, getnameinfo(). To use this API, a socket address structure is provided. The function prototype for getnameinfo() is the following:

```
int getnameinfo(

    IN const struct sockaddr FAR *sa,
```

IN socklen_t salen,

OUT char FAR *host,

IN size_t hostlen,

OUT char FAR *serv,

IN size_t servlen,

IN int flags

);

It contains the address and port in question. This can be either an IPv4 or IPv6 socket address structure because the length is also provided.

Additionally, buffers are provided to receive the node name and service name associated with that address, and the flags field can be used to change the default behavior of the API. The lengths of these buffers are provided in the call, and constants are defined (NI_MAXHOST, NI_MAXSERV) to aid the application in allocating buffers of the maximum size required.

The flags adjust the behavior as follows:

- NI_NOFQDN

  Setting the NI_NOFQDN flag results in returning only the node name (not the fully qualified domain name, or FQDN) for local hosts.

- NI_NUMERICHOST

  Setting this flag results in returning the numeric form of the host's address instead of its name.

- NI_NAMEREQD

  Setting the NI_NAMEREQD flag results in returning an error if the name cannot be located.

- NI_NUMERICSERV

  Setting NI_NUMERICSERV results in returning the numeric port number instead of the service name.

- NI_DGRAM

  Setting this flag specifies that the service is a datagram service, causing a search for a UDP service (instead of a TCP service).

USING GETADDRINFO

Here is an example of a client application using getaddrinfo() to connect to a specific server:

```
{/* Client Side...*/}

if (getaddrinfo(service_name, port, NULL, &ai) != 0) {/* Error Handling */}


conn_socket = socket(ai->ai_family, ai->ai_socktype,ai->ai_protocol);

if (conn_socket <0 ) {/* Error Handling */}


if (connect(conn_socket,ai->ai_addr,ai->ai_addrlen) == SOCKET_ERROR)

{/* Error Handling */}


freeaddrinfo(ai);
```

For an example of an application that checks each address returned, see the white paper titled "Adding IPv6 Capability to Windows Sockets Applications" in the \White_Papers folder on the companion CD-ROM.

Here is an example of the corresponding server application using getaddrinfo() to resolve the address information for the socket creation and bind calls:

```
{/* Server Side... */}

hints.ai_family = AF_INET6;

hints.ai_socktype = SOCK_STREAM;

hints.ai_flags = AI_NUMERICHOST | AI_PASSIVE;


retval = getaddrinfo(interface, port, &hints, &ai);

if (retval != 0) {/* Error Handling */}


listen_socket = socket(ai->ai_family, ai->ai_socktype, ai->ai_protocol);
```

```
if (listen_socket == INVALID_SOCKET){/* Error Handling */}


if (bind(listen_socket,ai->ai_addr,ai->ai_addrlen )== SOCKET_ERROR)

{/* Error Handling */}


freeaddrinfo(ai);
```

The interface parameter in this call could be NULL, or could be set to a numeric string.

ADDRESS CONVERSION FUNCTIONS

The inet_addr() and inet_ntoa() functions are provided to convert IPv4 addresses between binary and text formats. The IETF defined the similar functions, inet_pton() and inet_ntop(), to convert both IPv4 and IPv6 addresses. These functions contain an additional address family argument to make them protocol-independent.

Because getaddrinfo() and getnameinfo() provide the same functionality, the IETF is in the process of deprecating the inet_pton() and the inet_ntop() functions. As a result, the IPv6 protocol for Windows XP and the Windows .NET Server 2003 family does not support the inet_pton() and the inet_ntop() functions.

SOCKET OPTIONS

A new socket option level, IPPROTO_IPV6, has been defined for IPv6-specific socket options. Although an application can send multicast UDP packets by specifying a multicast address in sendto(), most of the new socket options currently defined for IPv6 are intended to adjust multicast behavior. New socket options are the following:

- IPV6_MULTICAST_IF

  This option sets the default interface to use for outgoing multicast traffic to the interface indicated by the index specified in the argument (0 indicates that the system chooses the interface).

- IPV6_MULTICAST_HOPS

  This option sets the hop limit for outgoing multicast packets based on the argument. Valid values are either 0 to 255 inclusive or -1 (to use the system default).

- IPV6_MULTICAST_LOOP

  This option controls whether outgoing multicast packets addressed to a group, of which the interface is a member, is looped back.

For reception of multicast traffic, new options are defined to join and leave multicast groups. These options take an argument of an *ipv6_mreq* structure:

```
struct ipv6_mreq {

    struct in6_addr ipv6mr_multiaddr;

    unsigned int ipv6mr_interface;

  };
```

This structure contains the multicast address of the group to be joined or left, and an interface index to use for this join or leave.

New multicast socket options are:

- IPV6_JOIN_GROUP

  This option is used to join the specified multicast group on the interface indicated (0 indicates that the system chooses the interface).

- IPV6_LEAVE_GROUP

  The IPV6_LEAVE_GROUP option is used to leave the specified group on the interface indicated.

In addition, another socket option, IPV6_UNICAST_HOPS, is defined to control the hop limit for outgoing unicast packets.

NEW MACROS

The additions to Windows Sockets that support IPv6 include the following set of macros to test addresses and determine whether they are special IPv6addresses:

- IN6_IS_ADDR_UNSPECIFIED
- IN6_IS_ADDR_LOOPBACK
- IN6_IS_ADDR_MULTICAST
- IN6_IS_ADDR_LINKLOCAL
- IN6_IS_ADDR_SITELOCAL
- IN6_IS_ADDR_V4MAPPED
- IN6_IS_ADDR_V4COMPAT
- IN6_IS_ADDR_MC_NODELOCAL

- IN6_IS_ADDR_MC_LINKLOCAL
- IN6_IS_ADDR_MC_SITELOCAL
- IN6_IS_ADDR_MC_ORGLOCAL
- IN6_IS_ADDR_MC_GLOBAL

The first seven macros return a true value if the address is of the specified type. The last five test the scope of a multicast address, returning a true value if the address is a multicast address of the specified scope and a false value if the address is either not a multicast address or is not of the specified scope.

The IN6_IS_ADDR_V4MAPPED macro can be use to determine whether the destination address for a socket is an IPv4 node. IPv4-mapped addresses are not supported by the IPv6 protocol for Windows XP and the Windows .NET Server 2003 family.

UNSUPPORTED APIS

The following are APIs that are currently specified in RFC 2553 or RFC 2292 that the IPv6 protocol for Windows XP and the Windows .NET Server 2003 family does not support:

- The APIs getipnodebyname(), getipnodebyaddr(), inet_pton(), and inet_ntop() are being deprecated by the IETF. They are redundant, as the functionality they provide is available with getaddrinfo() and getnameinfo().
- There is a set of name/interface index conversion functions that are not supported, but might be supported in future versions of Windows.
- The IETF is in the process of revising the advanced API specification (RFC 2292). The revised RFC will include a discussion of programming raw sockets and access to various options for IPv6 packets.

ANEXO D

BUILDING A LINUX IPV6 DNS SERVER

IPv6 is the next-generation protocol designed by the Internet Engineering Task Force (IETF) to replace IPv4, the current version of the Internet Protocol. IPv4 has been remarkably resilient. However, its initial design did not take into consideration several issues of importance today, such as a large address space, mobility, security, autoconfiguration and quality of service. To address these concerns, IETF has developed a suite of protocols and standards known as IPv6, which incorporates many of the concepts and proposed methods for updating IPv4. As a result, IPv6 fixes a number of problems in IPv4 and adds many improvements and features that cater to the future mobile Internet.

IPv6 is expected to replace IPv4 gradually, with the two coexisting for a number of years in a transition period. Servers will be dual stack, supporting both IPv4 and IPv6.

In this article, we look closely at IPv6 name resolution and provide a technical tutorial to help readers set up their own IPv6 Linux DNS servers to allow IPv6 name resolution using the latest version of BIND 9.x.

General Network Overview

In this section, we present a sample network scheme (Figure 1) with different IPv6 servers.

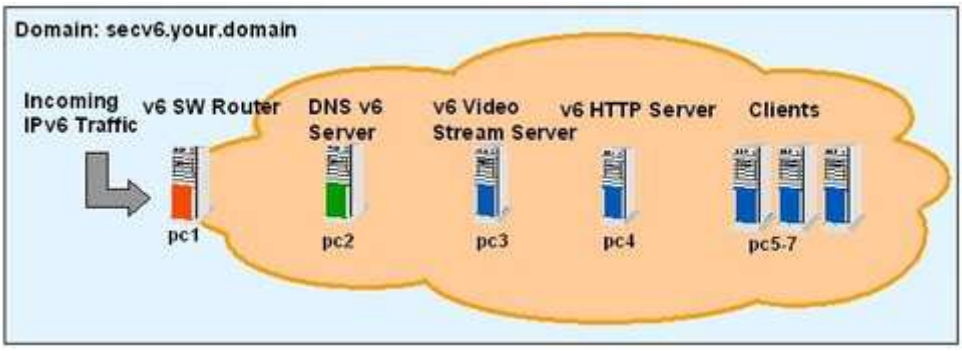

Figure 1. Sample Network Architecture

The following nodes are represented in this architecture:

- Routing server (pc1) acts as an IPv6 software router server and provides router advertisement for all IPv6 nodes.
- DNS IPv6 server (pc2) provides IPv6 name resolution.
- Two application servers, one provides video streaming (pc3) and the other is an Apache-based Web server (pc4).
- Client machines (pc5–7) used for testing purposes.

IPv6 Name Resolution

Domain names are a meaningful and easy-to-remember “handle” for Internet addresses. The domain name system (DNS) is the way that Internet domain names are located and translated into Internet protocol addresses. Because maintaining a central list of domain name/IP address correspondences is not practical, the lists of domain names and IP addresses are distributed throughout the Internet in a hierarchy of authority. Typically, a DNS server is within close geographic range of your access provider; this DNS server maps the domain names in DNS requests or forwards them to other servers on the Internet. For IPv6 DNS requests, both A6 and AAAA syntax are used to express IPv6 addresses.

AAAA resource record (called quad A record) is formatted as fixed-length data. With AAAA, we can define DNS records for IPv6 name resolution as follows, the same method as A records in IPv4:

```
$ORIGIN X.EXAMPLE.
N            AAAA 2345:00C1:CA11:0001:1234:5678:9ABC:DEF0
N            AAAA 2345:00D2:DA11:0001:1234:5678:9ABC:DEF0
N            AAAA 2345:000E:EB22:0001:1234:5678:9ABC:DEF0
```

An A6 resource record is formatted as variable-length data. With A6, it is possible to define an IPv6 address by using multiple DNS records. Here is an example taken from RFC 2874:

```
$ORIGIN X.EXAMPLE.
N            A6 64 ::1234:5678:9ABC:DEF0 SUBNET-1.IP6
SUBNET-1.IP6 A6 48 0:0:0:1::  IP6
IP6          A6 48 0::0       SUBSCRIBER-X.IP6.A.NET.
IP6          A6 48 0::0       SUBSCRIBER-X.IP6.B.NET.

SUBSCRIBER-X.IP6.A.NET. A6 40 0:0:0011:: A.NET.IP6.C.NET.
SUBSCRIBER-X.IP6.A.NET. A6 40 0:0:0011:: A.NET.IP6.D.NET.
SUBSCRIBER-X.IP6.B.NET. A6 40 0:0:0022:: B-NET.IP6.E.NET.
A.NET.IP6.C.NET. A6 28 0:0001:CA00:: C.NET.ALPHA-TLA.ORG.
A.NET.IP6.D.NET. A6 28 0:0002:DA00:: D.NET.ALPHA-TLA.ORG.
B-NET.IP6.E.NET. A6 32 0:0:EB00::    E.NET.ALPHA-TLA.ORG.
C.NET.ALPHA-TLA.ORG. A6 0 2345:00C0::
D.NET.ALPHA-TLA.ORG. A6 0 2345:00D0::
E.NET.ALPHA-TLA.ORG. A6 0 2345:000E::
```

If we translate the above code into AAAA records, it looks like:

```
$ORIGIN X.EXAMPLE.
N            AAAA 2345:00C1:CA11:0001:1234:5678:9ABC:DEF0
N            AAAA 2345:00D2:DA11:0001:1234:5678:9ABC:DEF0
N            AAAA 2345:000E:EB22:0001:1234:5678:9ABC:DEF0
```

Once IPv6 name resolution is configured, we can add domain name system (DNSSEC) to our DNS server. DNSSEC provides three distinct services: key distribution, data origin authentication and transaction and request authentication. The complete definition of DNSSEC is provided in RFC 2535.

Supporting IPv6 in the Kernel and in Network Binaries

An essential step prior to installing the IPv6-compliant BIND version is to enable IPv6 support in the kernel and for the networking binaries on the system supporting IPv6. We have covered this topic in a previous article, “Supporting IPv6 on a Linux Server Node”, in the August 2002 issue of *LJ* (http:www.linuxjournal.com/article/4763). After following the tutorial presented in that article, you should be ready to install the latest BIND version with IPv6 support.

BIND and IPv6 Support

The latest version of BIND is available from the Internet Software Consortium Web site (www.isc.org/products/BIND/BIND9.html). BIND version 9 is a major rewrite of nearly all aspects of the underlying BIND architecture. Many important features and enhancements were introduced in version 9; the most relevant to this article is the support for IPv6. BIND 9.x allows the DNS server to answer DNS queries on IPv6 sockets, provides support for IPv6 resource records (A6, DNAME and so on) and supports bitstring labels. In addition, BIND 9.x makes available an experimental IPv6 resolver library. Many other features are available, and you can read more about them from the BIND Web site.

Installing BIND 9.x

BIND 9.2.1 is the latest stable release available at the time of this writing. Our installation and configuration procedure follows this version. To install BIND, begin by downloading the latest BIND version into /usr/src, and then uncompress the package with:

```
% tar -xzf bind-9.2.1.tar.gz
% cd bind-9.2.1
```

Although IPv6 support is native to BIND, it must be specified explicitly when compiling. In addition, because we want to support DNSSEC, we need to compile BIND with crypto support. OpenSSL 0.9.5a or newer should be installed. Running the configuration script with the needed options looks like:

```
% ./configure -enable-ipv6 -with-openssl
```

Finally, compile and install the package as root with:

```
% make && make install
```

By default, the BIND 9 files are distributed in the filesystem. Configuration files are placed in /etc/named.conf; the binary “named” is in /usr/local/sbin and all other related configuration files go in /var/named.

Configuring IPv6 DNS and DNSSEC

DNS queries can be resolved in many different ways. For instance, a DNS server can use its cache to answer a query or contact other DNS servers on behalf of the client to resolve the name fully. When the DNS server receives a query, it first checks to see if it can answer it authoritatively, based on the resource record information contained in a locally configured zone on the server. If the queried name matches a corresponding resource record in the local zone information, the server answers authoritatively, using this information to resolve the queried name. For a complete DNS query process, there are four existing DNS zones:

1. Master: the server has the master copy of the zone data and provides authoritative answers for it.
2. Slave: a slave zone is a copy of a master zone. Each slave zone has a list of masters that it may query to receive updates to its copy of the zone. A slave, optionally, may keep a copy of the zone saved on disk to speed startups. A single master server can have any number of slaves in order to distribute load.
3. Stub: a stub zone is much like a slave zone and behaves similarly, but it replicates only the NS records of a master zone rather than the whole zone. Stub zones keep track of which DNS servers are authoritative for the organization. They directly contact the root DNS server to determine which servers are authoritative for which domain.
4. Forward: a forward zone directs all queries in the zone to other servers. As such, it acts as a caching DNS server for a network. Or it can provide Internet DNS services to a network behind a firewall that limits outside DNS queries, but obviously the forwarding DNS server must have DNS access to the Internet. This situation is similar to the global forwarding facility but allows per-zone selection of forwarders.

To map this to our network (Figure 1), we need to create a master server for our own domain, secv6.your.domain. Listing 1 provides a sample /etc/named.conf configuration. (The secret key is truncated to fit on a line.)

Listing 1. /etc/named.conf

```
options {
directory "/var/named";

// a caching only nameserver config
zone "." IN {
type hint;
file "named.ca";
};

// this defines the loopback name lookup
zone "localhost" IN {
type master;
file "master/localhost.zone";
allow-update { none; };
};

// this defines the loopback reverse name lookup
```

```
zone "0.0.127.in-addr.arpa" IN {
type master;
file "master/localhost.rev";
allow-update { none; };
};

// This defines the secv6 domain name lookup
// Secure (signed) zone file is
// secv6.your.domain.signed
// Regular zone file is secv6.your.domain
zone "secv6.your.domain" IN {
type master;
file "master/secv6.your.domain.signed";
// file "master/secv6.your.domain";
};

// this defines the secv6 domain reverse
// name lookup (AAAA)
zone "secv6.int" IN {
type master;
file "master/secv6.int";
};

// this defines the secv6 domain reverse
// name lookup (A6)
zone "secv6.arpa" IN {
type master;
file "master/secv6.rev";
};

// secret key truncated to fit
key "key" {
algorithm hmac-md5;
secret "HxbmAnSO0quVxcxBDjmAmjrmhgDUVFcFNcfmHC";
};
```

The next step is to define the configuration files that describe our domain. Notice that until now we have not touched on the specifics of IPv6. As for DNSSEC, the file /var/named/master/secv6.your.domain.signed is the domain file signed by the zone key of the DNS server. This is important to DNSSEC, because clients are able to authenticate all subsequent DNS requests. The DNS server zone key is different from the key in the configuration file; the details on how to generate a zone key are discussed later in the article.

The next file to edit is /var/named/master/secv6.your.domain. Our example (Listing 2) uses both AAAA and A6 formats. The $INCLUDE directive at the end includes the public portion of the zone key. Keep the private portion of the key private. The private key has `private` appended at the end, whereas `key` postfixes the public key. If you have any concerns regarding DNSSEC keys and their permissions, consult the BIND manual. In Listing 2, we display a typical IPv6 DNS domain configuration for secv6.your.domain.

Listing 2. /var/named/master/secv6.your.domain

```
$TTL 86400
$ORIGIN secv6.your.domain.
```

```
@ IN SOA secv6.your.domain. hostmaster.your.domain. (
2002011442 ; Serial number (yyyymmdd-num)
3H ; Refresh
15M ; Retry
1W ; Expire
1D ) ; Minimum
IN MX 10 noah.your.domain.
IN NS ns.secv6.your.domain.
$ORIGIN secv6.your.domain.
ns 1D IN AAAA fec0::1:250:b7ff:fe14:35d0
1D IN A6 0 fec0::1:250:b7ff:fe14:35d0
secv6.your.domain. 1D IN AAAA fec0::1:250:b7ff:fe14:35d0 1D IN A6 0
fec0::1:250:b7ff:fe14:35d0
pc2 1D IN AAAA fec0::1:250:b7ff:fe14:35d0  1D IN A6 0
fec0::1:250:b7ff:fe14:35d0
pc3 1D IN A6 0 fec0::1:250:b9ff:fe00:131   1D IN AAAA
fec0::1:250:b9ff:fe00:131
pc6 1D IN A6 0 fec0::1:250:b7ff:fe14:3617  1D IN AAAA
fec0::1:250:b7ff:fe14:3617
pc4 1D IN A6 0 fec0::1:250:b7ff:fe14:35c4  1D IN AAAA
fec0::1:250:b7ff:fe14:35c4
pc5 1D IN A6 0 fec0::1:250:b7ff:fe14:361b  1D IN AAAA
fec0::1:250:b7ff:fe14:361b
pc7 1D IN A6 0 fec0::1:250:b7ff:fe14:365a  1D IN AAAA
fec0::1:250:b7ff:fe14:365a
pc1 1D IN A6 0 fec0::1:250:b9ff:fe00:12e   1D IN AAAA
fec0::1:250:b9ff:fe00:12e
pc1 1D IN A6 0 fec0:0:0:1::1 1D IN AAAA fec0:0:0:1::1
$INCLUDE "/var/named/master/Ksecv6.your.domain.+003+27034.key"
```

For configuration files in /var/named/master, Hostmaster actually is the e-mail address of the administrator, where the first dot replaces the at symbol (@) because of syntax restrictions. In addition, the first number for the IN SOA structure at the beginning of Listing 2 is the serial number conventionally expressed as YYYYMMDDNN, where NN is a number incremented each time the DNS zone is updated.

Now, we discuss how to generate a zone key. The working directory for this step is important because the keys are placed there. We suggest placing the keys in /var/named/master. The following command generates a 768-bit DSA key for the zone:

```
% dnssec-keygen -a DSA -b 768 -n ZONE \
secv6.your.domain
```

By default, all zone keys that have an available private key are used to generate signatures. The keys must be either in the working directory or included in the zone file. The following command signs the secv6.your.domain zone, assuming it is in a file called /var/named/master/secv6.your.domain:

```
% dnssec-signzone -o secv6.your.domain \
secv6.your.domain
```

One output file is produced: /var/named/master/secv6.your.domain.signed. This file should be referenced by /etc/named.conf as the input file for the zone.

The remaining configuration files are localhost.zone (Listing 3), localhost.rev (Listing 4), secv6.rev (Listing 5) and secv6.int (Listing 6). The difference between reverse lookup zone files secv6.rev and secv6.int is that one can be specified using A6 strings (that do not need to be reversed in secv6.rev) and the other with reverse AAAA format addresses in secv6.int. For instance, ping6 can refer only to secv6.int domain because it does not support A6 format.

Listing 3. /var/named/master/localhost.zone

```
// localhost.zone  Allows for local communications
// using the loopback interface
$TTL 86400
$ORIGIN localhost.
@ 1D IN SOA @ root (
42 ; serial (d. adams)
3H ; refresh
15M ; retry
1W ; expire
1D ) ; minimum
1D IN NS @
1D IN A 127.0.0.1
```

Listing 4. /var/named/master/localhost.rev

```
// localhost.rev  Defines reverse DNS lookup on
// loopback interface
$TTL 86400
$ORIGIN 0.0.127.in-addr.arpa.
@ IN SOA 0.0.127.in-addr.arpa. hostmaster.secv6.your.domain. (
42 ; Serial number (d. adams)
3H ; Refresh
15M ; Retry
1W ; Expire
1D ) ; Minimum
NS ns.secv6.your.domain.
MX 10 noah.ip6.your.domain.
PTR localhost.
```

Listing 5. /var/named/master/secv6.rev

```
// secv6.rev  Defines reverse lookup for secv6
// domain in A6 format
$TTL 86400
$ORIGIN secv6.arpa.
@ IN SOA secv6.arpa. hostmaster.secv6.your.domain. (
2002011442 ; Serial number (yyyymmdd-num)
3H ; Refresh
15M ; Retry
1W ; Expire
1D ) ; Minimum
NS ns.secv6.your.domain.
MX 10 noah.your.domain.
; fec0:0:0:1::/64
$ORIGIN \[xfec0000000000001/64].secv6.arpa.
\[x0250b7fffe1435d0/64] 1D IN PTR pc2.secv6.your.domain.
\[x0250b9fffe000131/64] 1D IN PTR pc3.secv6.your.domain.
```

```
\[x0250b7fffe143617/64] 1D IN PTR pc6.secv6.your.domain.
\[x0250b7fffe1435c4/64] 1D IN PTR pc4.secv6.your.domain.
\[x0250b7fffe14361b/64] 1D IN PTR pc5.secv6.your.domain.
\[x0250b7fffe14365a/64] 1D IN PTR pc7.secv6.your.domain.
\[x0250b9fffe00012e/64] 1D IN PTR pc1.secv6.your.domain.
```

Listing 6. /var/named/master/secv6.int

```
// secv6.int  Defines reverse lookup for secv6
// domain in AAA format
$TTL 86400
$ORIGIN secv6.int.
@ IN SOA secv6.int. hostmaster.secv6.your.domain. (
2002011442 ; Serial number (yyyymmdd-num)
3H ; Refresh
15M ; Retry
1W ; Expire
1D ) ; Minimum
NS ns.secv6.your.domain.
MX 10 noah.your.domain.
; fec0:0:0:1::/64
$ORIGIN 1.0.0.0.0.0.0.0.0.0.0.0.c.e.f.secv6.int.
0.d.5.3.4.1.e.f.f.f.7.b.0.5.2.0 IN PTR pc2.secv6.your.domain.
e.2.1.0.0.0.e.f.f.f.9.b.0.5.2.0 IN PTR pc1.secv6.your.domain.
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN PTR pc1.secv6.your.domain.
1.3.1.0.0.0.e.f.f.f.9.b.0.5.2.0 IN PTR pc3.secv6.your.domain.
7.1.6.3.4.1.e.f.f.f.7.b.0.5.2.0 IN PTR pc6.secv6.your.domain.
4.c.5.3.4.1.e.f.f.f.7.b.0.5.2.0 IN PTR pc4.secv6.your.domain.
b.1.6.3.4.1.e.f.f.f.7.b.0.5.2.0 IN PTR pc5.secv6.your.domain.
```

Starting DNS Dæmon

Once the installation and configuration steps are complete, you are ready to start the DNS dæmon on pc2. Named uses /etc/named.conf by default, although you can specify a different configuration file with the -c option if you want. Depending on where you installed the dæmon, enter:

```
pc2% /usr/local/sbin/named
```

One additional configuration step is needed on the machines within the IPv6 network: update /etc/resolv.conf (Listing 7) to contain the DNS server's IP address. It is important that the IP address is included and not the hostname of the DNS server, because this file is where the system looks to find the address of the DNS. In other words, if you specified the hostname of the DNS server here, how would the system know what IP address corresponds to the DNS' hostname?

Listing 7. /etc/resolv.conf on Client Machines

```
# To enable secv6 domain, start named on pc2
# and use this file as /etc/resolv.conf
search secv6.your.domain
nameserver fec0::1:250:b7ff:fe14:35d0
```

Testing the Setup

We use two simple methods of testing the setup. The first verifies that A6 addresses are enabled in the DNS server, and the second verifies that AAAA addresses are supported by the DNS server. The tests were performed on pc2. We present only the meaningful output here; otherwise the listing would be too long. For the first example, we use the DNS lookup utility **dig** to perform a lookup on secv6 domain in A6 format (Listing 8). We then perform a lookup in AAAA format (Listing 9). In both cases, we are not specifying an address to look up, thus our use of 0.0.0.0.

Listing 8. A6 DNS Query

```
pc2% dig 0.0.0.0 secv6.your.domain a6
; <<>> DiG 9.1.0 <<>> 0.0.0.0 secv6.your.domain A6
[...]
;secv6.your.domain. IN A6
;; ANSWER SECTION:
secv6.your.domain. 86400 IN A6 0 fec0::1:250:b7ff:fe14:35d0
;; AUTHORITY SECTION:
secv6.your.domain. 86400 IN NS ns.secv6.your.domain.
;; ADDITIONAL SECTION:
ns.secv6.your.domain. 86400 IN A6 0 fec0::1:250:b7ff:fe14:35d0
ns.secv6.your.domain. 86400 IN AAAA fec0::1:250:b7ff:fe14:35d0
```

Listing 9. AAAA DNS Query

```
pc2% dig 0.0.0.0 secv6.your.domain aaaa
; <<>> DiG 9.1.0 <<>> 0.0.0.0 secv6.your.domain AAAA
[...]
;secv6.your.domain. IN AAAA
;; ANSWER SECTION:
secv6.your.domain. 86400 IN AAAA fec0::1:250:b7ff:fe14:35d0
;; AUTHORITY SECTION:
secv6.your.domain. 86400 IN NS ns.secv6.your.domain.
;; ADDITIONAL SECTION:
ns.secv6.your.domain. 86400 IN A6 0 fec0::1:250:b7ff:fe14:35d0
ns.secv6.your.domain. 86400 IN AAAA fec0::1:250:b7ff:fe14:35d0
```

For our second test, we include samples of an SSH session connection, first using an IPv6 address and then using an IPv6 hostname.

Sample Server Applications Using IPv6

In our IPv6 network, we presented two application servers: Apache as a Web server and VideoLan for video streaming. To test IPv6 name resolution when streaming a video, a user on client node pc5 accesses the video-streaming server on pc3. The video server is on pc3 (fec0::1:250:b7ff:fe14:5768), and the video client is on pc5 (fec0::1:250:b7ff:fe50:7c). Sniffing the network communications on pc5 with **tcpdump**, we captured packets from the video stream. Here is a portion of the trace:

```
% tcpdump ip6    # only trace IPv6 traffic, must be run as root or setuid
root
[snip...]
02:09:26.716040 fec0::1:250:b7ff:fe14:5768.32769 >
fec0::1:250:b7ff:fe50:7c.1234: udp 1316
02:09:26.735805 fec0::1:250:b7ff:fe14:5768.32769 >
fec0::1:250:b7ff:fe50:7c.1234: udp 1316
02:09:26.735971 fec0::1:250:b7ff:fe14:5768.32769 >
fec0::1:250:b7ff:fe50:7c.1234: udp 1316
02:09:26.736082 fec0::1:250:b7ff:fe14:5768.32769 >
fec0::1:250:b7ff:fe50:7c.1234: udp 1316
02:09:26.755810 fec0::1:250:b7ff:fe14:5768.32769 >
fec0::1:250:b7ff:fe50:7c.1234: udp 1316
02:09:26.755935 fec0::1:250:b7ff:fe14:5768.32769 >
fec0::1:250:b7ff:fe50:7c.1234: udp 1316
02:09:26.775787 fec0::1:250:b7ff:fe14:5768.32769 >
fec0::1:250:b7ff:fe50:7c.1234: udp 1316
```