UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES
CUAUTITLÁN

"PROPUESTA DE IMPLEMENTACIÓN DE IPv6 EN FES CUAUTITLÁN CAMPO 4"

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO MECÁNICO ELECTRICISTA

P R E S E N T A:

GREGORIO GABRIEL LEMUS RAYA

ASESOR: ING. JESÚS MOISÉS HERNÁNDEZ DUARTE

CUAUTITLAN IZCALLI, EDO. DE MEX. 2004

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLAN
UNIDAD DE LA ADMINISTRACION ESCOLAR
DEPARTAMENTO DE EXAMENES PROFESIONALES

ASUNTO: VOTOS APROBATORIOS

U. N. A. M.
FACULTAD DE ESTUDIOS
SUPERIORES-CUAUTITLAN

DEPARTAMENTO DE

DR. JUAN ANTONIO MONTARAZ CRESPO
DIRECTOR DE LA FES CUAUTITLAN
P R E S E N T E

ATN Q. Ma. del Carmen García Mijares
Jefe del Departamento de Exámenes
Profesionales de la FES Cuautitlán

Con base en el art. 28 del Reglamento General de Exámenes, nos permitimos comunicar a usted que revisamos la TESIS:
Propuesta de implementación de IPv6 en FES-Cuautitlán campo 4.

que presenta el pasante: Gregorio Gabriel Lemus Raya
con número de cuenta: 9124815-6 para obtener el titulo de
Ingeniero Mecánico Electricista

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el EXAMEN PROFESIONAL correspondiente, otorgamos nuestro VOTO APROBATORIO.

A T E N T A M E N T E
"POR MI RAZA HABLARA EL ESPIRITU"
Cuautitlán Izcalli, Méx. a 14 de abril de 2004

PRESIDENTE Ing. José Luis Cruz Gutiérrez
VOCAL Ing. Jesús Moisés Hernández Duarte
SECRETARIO Ing. Victor Manuel Cuevas Rodríguez
PRIMER SUPLENTE Ing. Angel Hilario García Bacho
SEGUNDO SUPLENTE M.A.I. Pedro Guzmán Tinajero

DEDICATORIAS

Este trabajo está dedicado a mis papás:

Luis Lemus Martínez

Y

María Guadalupe Raya

A ellos, por haberme infundado principios y valores que fueron la base para mi formación personal y profesional que con sus consejos, comprensión y desvelos me ayudaron llegar a la meta.

Ustedes me han enseñado las bases fundamentales para seguir adelante: la fe, la dedicación, la honestidad y el trabajo constante.

Hoy se pueden sentir satisfechos de que sus esfuerzos no fueron en vano.

"Mis triunfos son de Ustedes y para Ustedes"

GRACIAS POR TODO LO QUE ME HAN BRINDADO!!!

Gregorio Gabriel Lemus Raya

AGRADECIMIENTOS

A Dios,

Por darme la luz que ha iluminado mi camino.
Por estar siempre cerca de mi cuando te he necesitado.
Por darme la fortaleza para concluir mi carrera y por que se me has bendecido con todo lo que me has dado.

A mis hermanos,
Alex, Toño, Paty

Por el cariño y apoyo incondicional que me han brindado y sobre todo por que juntos en la distancia seguiremos siendo una gran familia.

A mis sobrinas(os),
Ireri, Yuri, Ayleen y Luis Ruben.

Por su amor y cariño sincero.

AGRADECIMIENTOS

A mis amigos,
Carlos, Jorge, Israel, Agustín, Xoemi

Con ustedes he compartido una de las etapas más hermosas de mi vida: la de ser estudiante, y con ello ha quedado un mundo lleno de recuerdos maravillosos y gracias por estar conmigo en las buenas y en las malas.

A mis compañeros de la DGSCA,
Hugol, Charly, Hans, Alfred, Felpa, Alex, Oscar, Edgar, Israel, Yola, Chio, Pao, Lupita, a todos los integrantes del TAC, XIC, XOC.

Por brindarme su confianza, su amistad y sobre todo por compartir sus conocimientos y su experiencia conmigo

A mi Asesor,
Ing. Jesús Moisés Hernández Duarte

Por brindarme su apoyo desde inicios hasta la culminación de mi carrera profesional.
Por sus consejos que fueron parte importante en este trabajo de tesis.

AGRADECIMIENTOS

A la UNAM,

Por haberme dado la oportunidad de formar parte de su comunidad y darme el privilegio de tener una formación profesional.
Siempre llevaré en mí corazón con orgullo el tener la piel dorada y la sangre azul que sólo la UNAM puede brindar.... Goya!!!.

A la FES-C,

Un agradecimiento especial a la FESC porque pase años que fueron de grandes aventuras y travesías y nunca olvidaré los días lluviosos, con sol o frío.
Por enseñarme que siempre hay más de un camino para llegar a nuestro destino, no importando que tan difícil sea.

"El éxito se alcanza convirtiendo cada paso en una meta y cada meta en un paso." -C.C. Cortez.

"Llegará un momento en que creas que todo ha terminado...Ese será sólo el principio" - Anónimo

Gregorio Gabriel Lemus Raya

Índice

Objetivo General

- Con el presente trabajo de investigación se propone, implementar y probar el protocolo IPv6 en FES Cuautitlán campo 4.

Objetivos específicos

- Se diseñará un esquema para la implementación del protocolo IPv6 en la FES-Cuautitlán campo 4.
- Se configurará un túnel para la conexión con IPv6.

Introducción

Internet está experimentando un fenómeno de crecimiento que ha superado por mucho las expectativas de quienes diseñaron el Protocolo de Internet (IP) hace más de 20 años. Internet es una red de computadoras alrededor de todo el mundo, que comparten información unas con otras.

Los Protocolos en un entorno de red definen las reglas y los procedimientos para transmitir datos. Enviar datos a través de la red conlleva a una serie de pasos que deben realizarse de una forma coherente con el fin de que se realice la comunicación. Estos protocolos están organizados en capas basados en el modelo OSI.

El conjunto de protocolos llamado TCP/IP hace posible el funcionamiento de Internet, es decir es su base fundamental. A medida que Internet crece, TCP/IP se desarrolla paralelamente incorporando nuevas características.

Cuando se diseñó el protocolo IP versión 4 (IPv4) se planeó un espacio de direcciones de 32 bits. El espacio de direcciones que ha sido asignado es cada vez es más escaso y se torna más difícil para las organizaciones que desean conectarse a Internet obtener éstas direcciones.

En los últimos años la Fuerza de Tareas de Ingeniería de Internet (IETF por sus siglas en inglés) ha estado trabajando en soluciones a corto y largo plazo sobre este problema de direccionamiento. La solución a largo plazo es IPv6, una nueva versión para el protocolo IP que ha sido diseñado con nuevas e importantes características que permitirán superar las limitaciones del protocolo IPv4.

La innovación más importante en IPv6 es un espacio de direcciones mucho más grande. Las direcciones tienen una longitud de 128 bits lo que proporciona un espacio de direcciones prácticamente infinito.

Además del gran espacio de direcciones, IPv6 cuenta con nuevas características necesarias para la evolución de Internet tales como:

- Transición gradual de IPv4 a IPv6.
- Soporte a Multidifusión IP como una parte integral de IPv6.
- Soporte a Seguridad en nivel IP (autenticación y cifrado).
- Soporte a la Autoconfiguración de computadoras y ruteadores.
- Soporte para nodos móviles en el nivel de IP.
- Manejo eficiente de paquetes difiriendo los requerimientos de calidad de servicio (QoS).

Éstas características harán posible la coexistencia de redes de nueva generación más grandes, eficientes y seguras.

Se han creado varias iniciativas de investigación para promover y facilitar el conocimiento de IPv6 a nivel internacional. Algunas de estas redes de investigación como el 6BONE funciona bajo asignación de prueba del direccionamiento IPv6.

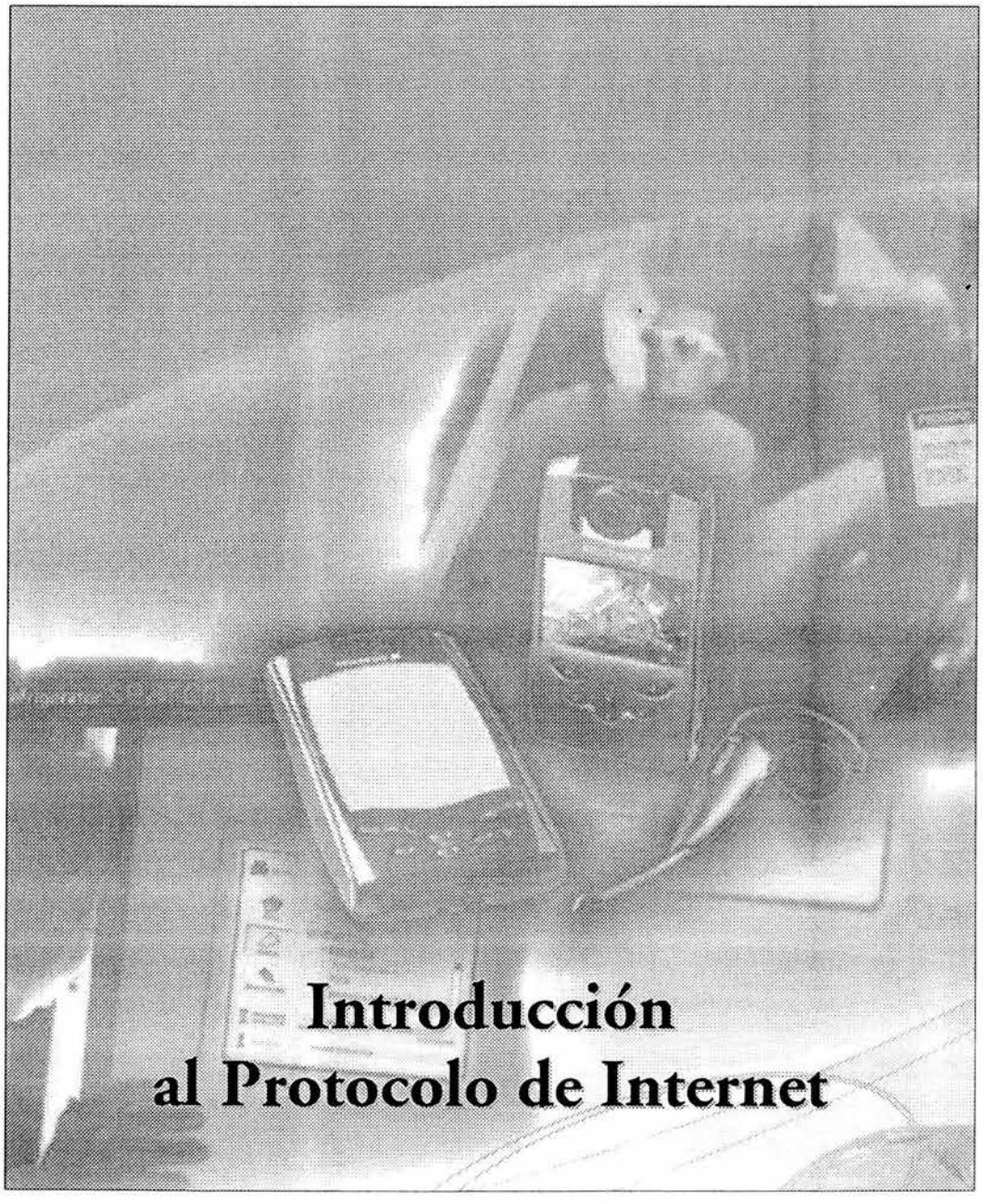
La Universidad Nacional Autónoma de México ha tenido el liderazgo en México desde 1999 sobre el proyecto de IPv6. También a obtenido un rango de direcciones tipo TLA (Top Level Agregation) tanto de pruebas como de producción. La UNAM puede delegar direcciones y configurar túneles a instituciones en México y en el mundo interesadas en realizar pruebas con IPv6.

Debido a las nuevas tendencias tecnológicas que vienen surgiendo es menester implementar este nuevo Protocolo de Internet en FES-Cuautitlán y de esta manera impulsar el desarrollo de las tecnologías emergentes hoy en día.

El trabajo escrito que se desarrolla a continuación pretende dar a conocer las características del protocolo IPv6, sus ventajas y desventajas, las diferencias con IPv4, los mecanismos de transición de IPv4 a IPv6, y ver el desarrollo que tiene a nivel internacional, así como el proceso de implementación dentro de la FES-C.

Se diseñará un esquema para la implementación y un esquema de direccionamiento del protocolo IPv6. Finalmente se realizarán pruebas con diferentes sistemas operativos y se utilizarán herramientas de red para la comprobación del stack IPv6.

Capítulo 1



Introducción al Protocolo de Internet

CAPITULO 1

Introducción al Protocolo de Internet (IP)

1.1 Modelo de Referencia OSI

El proceso de llevar datos de un equipo a otro es complicado, por ello fue necesario dividir este proceso en capas. De aquí surge la idea de tener protocolos estándar que permitan la comunicación entre hardware y software de distintos fabricantes.

En 1978 la Organización Internacional de Estándares (ISO) publicó un conjunto de especificaciones que describían una arquitectura de red para conectar diferentes dispositivos. En 1984, la ISO publicó una revisión de este modelo y lo llamó el modelo de referencia para Interconexión de Sistemas Abiertos (OSI)[1].

Este modelo es la referencia más conocida y más ampliamente usada para describir los entornos de red. Los fabricantes diseñan sus productos de red basados en las especificaciones del modelo OSI. Éste proporciona una descripción de cómo funcionan juntos por niveles el hardware y el software de red para hacer posible las comunicaciones.

El modelo OSI, es una arquitectura que se divide en varias capas, para ser específicos en siete y apiladas una sobre otra. Cada una utiliza protocolos de comunicación para realizar una función bien definida. Cómo se muestra en la figura 1.1.

Cada capa proporciona algún servicio o acción que prepara los datos para entregarlos a la siguiente capa, o sea todas las peticiones pasan de una capa a otra, a través de límites llamados interfaces. Cada nivel se construye sobre los estándares y actividades del nivel inferior. Las siete capas del modelo OSI se pueden dividir en dos categorías: capas superiores y capas inferiores.

Las capas superiores tiene que ver con la aplicación, que es la más cercana al usuario, definen cómo tienen acceso las aplicaciones a los servicios de comunicación. Por ejemplo, el nivel de sesión debe comunicarse y funcionar con los niveles de presentación y de transporte; cuanto mayor sea el nivel más compleja será su tarea.

Las capas inferiores del modelo OSI manejan la trasferencia de datos. Las capas física y de enlace de datos se encuentran implementadas en hardware y software. Por ejemplo definen el medio físico de la red y las tareas relacionadas, como por ejemplo, colocar los bits de datos en las tarjetas adaptadoras y el cable de red[2].

---

[1] Fundamentos de Redes, Editorial Microsoft Press, p. 167.
[2] Ford Merilee, Tecnologías de interconectividad de redes, p. 7

En la figura 1.1 se representa la arquitectura de niveles del modelo OSI.

Figura 1.1 El modelo OSI

Para que dos computadoras puedan comunicarse en una red tienen que operar bajo el mismo modelo teórico, tal como lo es el modelo OSI. La información que viaja por una red suele llamarse dato, paquete, o paquete de datos. Se llaman paquetes a las pequeñas unidades de fácil transmisión en las que se dividen los datos para su envío por la red. Podemos emplear el término *tramas* cuando hablamos de los paquetes. Además, se llama segmento a la unidad de datos que transmite el protocolo para el control de la transmisión (Transport Control Protocol, TCP). La dirección de origen de un paquete especifica la identidad de la computadora que lo envía. La dirección destino especifica la identidad de la computadora que lo recibe[3]. Ver Figura 1.2.

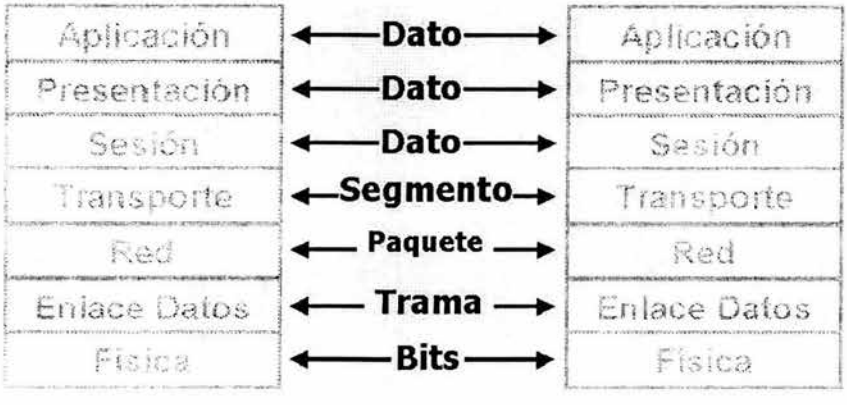


Figura 1.2 Transferencia de datos por los diferentes niveles del modelo OSI

[3] Guía del primer año, Cisco Systems, p 47.

En conclusión empleamos el término paquete para referirnos a los fragmentos de los datos en general. Un paquete de datos es una unidad de agrupación lógica de información; incluye la información sobre el origen, además de otros elementos que son necesarios para hacer posible una comunicación fiable con el dispositivo destino.

La ventaja de dividir los datos en paquetes son varias, por ejemplo, no utilizarían mucho ancho de banda, si se pierde un paquete solo se transmite una pequeña cantidad de datos en lugar de todo el archivo. Cada paquete puede tomar una ruta diferente para llegar a su destino o sea que si una ruta se congestiona o pierde velocidad los paquetes que siguen podrán tomar una ruta mejor.

Pongamos un ejemplo, supongamos que un equipo A envía una información al equipo B. La información del equipo A comienza en la parte superior de la arquitectura que es la capa de aplicación y debe pasar a través de todos los niveles inferiores. Estos datos se pasan a la capa de presentación, que los transforma y añade información perteneciente a este nivel y continúa bajando por cada una de las capas hasta llegar a la capa física, que envía la información a través del medio físico de comunicación: par trenzado, cable coaxial, fibra óptica, entre otros. Cuando la información llega al otro extremo (equipo B), el nivel físico recibe la información, la decodifica y la pasa a la capa de enlace. Este último quita la información de la capa de enlace y la traslada a la capa de red para extraer la información relacionada con esta capa y el resto de la información se envía hacia niveles superiores de la arquitectura, y así sucesivamente, hasta que la información llega a la capa de aplicación del equipo B tal y como fue enviada por la capa de aplicación del equipo A, como se muestra en la figura 1.3.

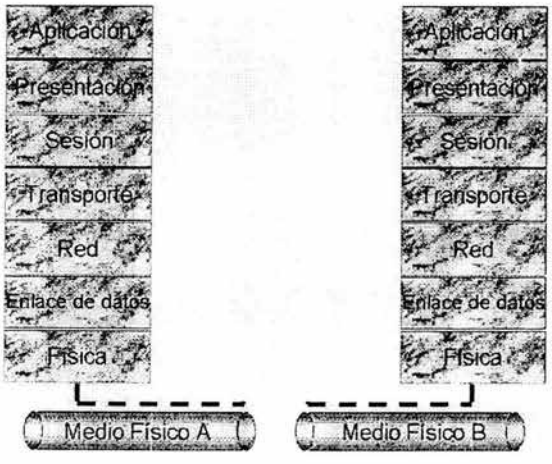


Figura 1.3 Transmisión y recepción de datos entre dos equipos

Cada una de las capas del modelo OSI actúa como módulo independiente, realizando un tipo de función específico y tiene su propio formato de instrucciones de comunicación, en forma de protocolos. Ver figura 1.4.

Figura 1.4 El modelo OSI define funciones para cada capa

La información procedente de una capa que se transfiere a la siguiente se llama unidad de datos de protocolos (Protocol Data Unit, PDU). Se va añadiendo a la PDU una información de control a medida que la información pasa de capa. Cuando la PDU está preparada para pasar a la capa siguiente se añaden las instrucciones necesarias para transferirla a esa capa.

Cuando la siguiente capa recibe la PDU, se extrae la información de control y las instrucciones de transferencia. El paquete resultante se llama unidad de datos de servicio (Service Data Unit, SDU). Como la SDU se desplaza por las capas, cada una de ellas añade su propia información de control. De este modo, la trama va aumentando de tamaño según se desplaza por las capas[4].

1.1.1 Encapsulamiento

La información que se envía por la red debe empaquetarse primero mediante un proceso llamado encapsulación. La encapsulación envuelve los datos con la información de protocolo necesaria antes de enviarla por la red. Como se muestra en la figura 1.5

[4] Palmer Michael, Redes de computadoras, p. 66.

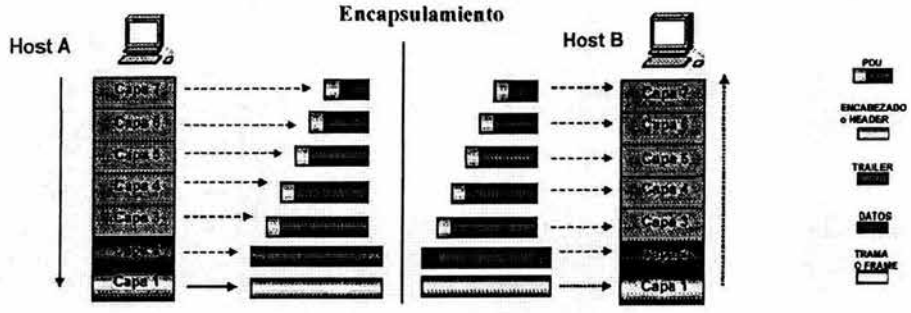


Figura 1.5 Proceso de encapsulación

Los datos, con forma de señales eléctricas, deben viajar por el cable hasta la computadora destino y, después, convertirse en su formado original para que puedan ser leídos por el destinatario.

Las redes deben realizar los siguientes pasos de conversión para encapsular los datos:

- *Construir los datos.* Cuando un usuario envía un correo electrónico, sus caracteres alfanuméricos se convierten en datos para que puedan viajar por la red.

- *Empaquetar los datos para el transporte de extremo a extremo.* Los datos se empaquetan para el transporte por la red. Usando segmentos, la función de transporte asegura que los equipos de ambos extremos del sistema puedan comunicarse con total fiabilidad.

- *Añadir la dirección de red a la cabecera.* Los datos se colocan en un paquete, o datagrama, que contiene una cabecera de red con direcciones lógicas de origen y destino. Dichas direcciones ayudan a los dispositivos de red a enviar paquetes a través de la red a lo largo de una ruta seleccionada dinámicamente.

- *Agregar la dirección local a la cabecera de enlace de datos.* Cada dispositivo de la red debe colocar el paquete en una trama. La trama incluye una cabecera con la dirección física del siguiente dispositivo conectado directamente en la misma ruta.

- *Convertir los bits para la transmisión.* La trama se debe convertir en un modelo de 1 y 0 (bits) para su transmisión por el medio físico. Una función de cronometraje permite a los dispositivos distinguir a los bits mientras viajan por el medio de transmisión.

Cuando el dispositivo remoto recibe una secuencia de bits, la pasa por la capa de enlace de datos para manipular las tramas. Cuando la capa de enlace de datos recibe la trama, hace lo siguiente:

- Lee la dirección física y otras informaciones de control que proporciona la capa de enlace de datos conectada directamente.
- Interpreta la información de control de la trama, creando un datagrama.
- Pasa el datagrama a la siguiente capa, siguiendo las instrucciones que aparecen en la zona de control de la trama.

1.1.2 Funciones de las capas del modelo OSI

Capa 1 Física

Es responsable del transporte de bits (ceros y unos). Dependiendo del tipo de *enlace físico* los bits se representan de una manera en que puedan ser transportados a través del medio. También define características como los niveles de voltaje, tiempo de duración de los pulsos, velocidades de transmisión, distancias máximas de conexión, el número de pines que tiene el conector y la función de cada pin, la forma de establecer la conexión inicial y de interrumpirla. El cableado, el equipo de red y el diseño físico de la red también forman parte de la capa física.

La capa física se ve afectada por los problemas físicos de la red, por ejemplo, si existe voltaje en un cable o si existen interferencias eléctricas o electromagnéticas. Las interferencias se producen por la proximidad de motores eléctricos, líneas de alta tensión, alumbrados y otros dispositivos eléctricos.

Los protocolos desarrollados por los comités de estándares incluyen las especificaciones del cable y las limitaciones en distancias del cable para la comunicación, por mencionar algunas, Ethernet, FDDI.

Capa 2 Enlace de datos

En la capa de enlace de datos permite que los datos se codifiquen como señales eléctricas en el nodo de transmisión, se decodifiquen en el nodo de recepción y se verifiquen para comprobar los errores. Entre las funciones que se definen en esta capa están la de asegurar que la información sea transmitida sin errores entre nodos, direccionamiento físico, topología de red, corrección y notificación de errores, secuencias de tramas y control de flujo.

Como la capa física rápidamente acepta y retransmite un flujo de bits sin tener en cuenta su significado o estructura, recae sobre la capa de enlace de datos la creación o los reconocimientos de los límites de la trama. Además resuelve los problemas de daño, pérdida o duplicidad de datos y participa en la regulación de flujo, por ejemplo, se evita que un transmisor muy rápido sature con datos a un

receptor muy lento; es responsable de conseguir una transmisión fiable entre ambos extremos de una comunicación cuyo medio físico puede presentar problemas de ruido y de interferencias en la transmisión.

El direccionamiento físico al contrario de el direccionamiento de red, indica cómo se asignan direcciones a los dispositivos en el nivel de enlace. La topología de la red indica el modelo de conexión de los dispositivos físicos, por ejemplo, en bus o en anillo. La notificación de errores avisa a los niveles superiores de que ha ocurrido un error en la comunicación y la corrección de errores permite detectar y eliminar errores que se hayan producido en la transmisión y si es necesario la capa pedirá una transmisión de datos, trama por trama, desde el nodo que los envía, mediante la asignación de número de secuencia a las tramas se pueden ordenar las que no llegan en el orden esperado. El control de flujo vigila que la velocidad de la comunicación sea la apropiada y no desborde al receptor de la misma.

En esta capa opera el Ethernet de Acceso Múltiple por Detección de Portadora con Detección de Colisiones (Carrier Sense Multiple Access Collision Detect, CSMA/CD) con el fin de determinar qué dispositivos deben transmitir en un momento dado para evitar colisiones, por ejemplo, si dos o más dispositivos tratan de transmitir señales al mismo tiempo, se producirá una colisión, por lo que, el CSMA/CD indicará al dispositivo que espere un periodo de tiempo antes de transmitir otra señal, con el fin de evitar colisiones.

El Instituto de Ingenieros en Electrónica y Electricidad (Institute of Electrical and Electronics Engineers, IEEE) divide la capa de enlace en dos subcapas:

a) Control de enlace lógico (Logical Link Control, LLC).- La subcapa LLC asegura que las transmisiones sean fiables, iniciando la comunicación entre los nodos y vigilando para que no se produzcan cortes en la conexión.

b) Control de acceso al medio (Medium Access Control, MAC).- La subcapa MAC se encarga de examinar la información de dirección que contiene cada una de las tramas. Otra de las funciones de la capa MAC es la de controlar cómo se comparte el medio de transmisión cuando hay más de un dispositivo en la misma red[5].

Capa 3 Red

Es la encargada de que los datos sean enviados a su correcto destino, determinando la ruta de transmisión. La unidad de transmisión de datos en esta capa es el paquete de datos. Entre sus funciones se encuentran las de direccionamiento lógico, control de congestión, conexión y desconexión de redes, control de flujo, detección de errores, entre otras.

[5] Ford Merilee, Op. Cit., p. 14

La capa de red soporta servicios orientados y no orientados a la conexión de los protocolos de las capas superiores. Los protocolos de la cada de red son de hecho *protocolos de ruteo*, sin embargo también otro tipo de protocolos están implementados en esta capa.

En muchas ocasiones se introduce una función de contabilidad en la capa de red, el software deberá saber cuantos paquetes o bits se enviaron a cada cliente con objeto de producir información de facturación.

Capa 4 Transporte

Su función principal consiste en aceptar los datos de la capa de sesión, dividirlos siempre que sean necesarios en unidades más pequeñas (la capa de red generalmente pone un límite en el tamaño de los mensajes que aceptan), pasarlos a la capa de red y asegurar que todos lleguen correctamente a su destino.

Entre sus funciones están las de control de flujo (que el origen no envíe más datos de los que pude aceptar el destino), multiplexación (enviar datos de varias aplicaciones a través del mismo enlace), gestión de circuitos virtuales y comprobación y recuperación de errores; también cuando se realiza una transmisión, el nodo de recepción puede enviar un acuse de recibo, que a veces se llama *"acknowledgment"* para indicar que los datos se han recibido. Es capaz de solicitar el establecimiento de un nuevo enlace en el caso de que falle un enlace de red.

Capa 5 Sesión

La capa de sesión es responsable de la continuidad de la conexión o sesión entre dos nodos. Establece la conexión y asegura que éste se mantenga mientras dure la sesión de comunicación. Además permite que los usuarios de diferentes computadoras puedan establecer sesiones entre ellos.

Uno de los servicios de la capa de sesión consiste en la realización de control de dialogo. Las sesiones permiten que el tráfico vaya en las mismas direcciones al mismo tiempo, o bien en una sola dirección en un instante dado.

Otro de los servicios de la capa de sesión es la sincronización, esta capa proporciona una forma de insertar puntos de verificación en el flujo de datos con el objeto de que solamente tengan que retransmitirse los datos que se encuentran enseguida del último punto de verificación cuando se reanude el servicio después de una caída de red.

Capa 6 Presentación

Esta capa es responsable de convertir los datos transmitidos a una forma inteligible. Proporciona un conjunto de funciones de codificación y conversión de

los datos; mediante de estas funciones se asegura que el sistema destino sea capaz de entender la información enviada independientemente de la codificación que utilicen; para los caracteres, por ejemplo, entre ASCII y EBCDIC o el uso de formatos estándar de vídeo, sonido, e imagen como, MPEG, JPEG, entre otros.

Entre sus funciones pueden citarse la conversión de formatos de representación de caracteres, cifrado, compresión de datos la cual elimina los espacios en blanco y los compacta para que los datos a enviar sean mucho más pequeños; los datos se descomprimirán en la capa de presentación del nodo receptor. Después de pasar este nivel los datos recibidos están disponibles en una forma en que la computadora entenderá.

Esta capa está relacionada también con otros aspectos de datos como la reducción del número de bits que tienen que transmitirse y la criptografía que se necesita utilizar frecuentemente por razones de privacidad y autenticación. La encriptación de los datos supone la codificación de los datos para que no puedan leerlos los usuarios no autorizados.

Capa 7 Aplicación

Ésta es la capa más cercana al usuario final, lo cual significa que tanto la capa de aplicación como el usuario interactúan de manera directa con la aplicación de software. Por ejemplo, correo electrónico (e-mail), transferencia de archivos (File Transfer Protocol, FTP), terminal virtual (telnet), transferencia de páginas web (Hipertex Transfer Protocol, HTTP) entre otros.

Las funciones de la capa de aplicación incluyen la identificación de socios de comunicación, la determinación de la disponibilidad de recursos y la sincronización de la comunicación.

Al identificar socios de comunicación, la capa de aplicación determina su identidad y disponibilidad para una aplicación que debe trasmitir datos y la capa de aplicación debe decidir si hay suficientes recursos en la red para la comunicación que se está solicitando. Al sincronizar la comunicación, toda comunicación entre aplicaciones requiere cooperación, y ésta es administrada por la capa de aplicación.

En la tabla 1.1 se explica cada una de las capas con las principales funciones y algunos de los protocolos que utiliza respectivamente.

Capa	Función	Protocolo
Aplicación	Interfaz de Usuario.	Telnet, HTTP, FTP, e-mail, SMTP, SNMP,
Presentación	✓ Presentación de los datos. ✓ Procesos especiales, como el	ASCII, EBCDIC, JPEG, GIF,

	cifrado, encriptación y compresión de datos.	MPEG, MP3
Sesión	✓ Mantiene separados los datos de distintas aplicaciones. ✓ Inicia, mantiene y termina una conexión o sesión.	Sistema operativo Programación de acceso a la aplicación
Transporte	✓ Asegura la fiabilidad de la transmisión de paquetes desde un nodo a otro nodo. ✓ Asegura que los paquetes se reciban en el mismo orden en que se enviaron. ✓ Monitorea los errores de transmisión de paquetes y reenvía los paquetes erróneos.	TCP, UDP
Red	✓ Proporciona el direccionamiento lógico que los *routers* utilizan para determinar la ruta. ✓ Establece circuitos virtuales. ✓ Enruta los paquetes hacia otras redes y reordena los paquetes cuando sea necesario.	IP, IPX, RIP, IGRP, OSPF
Enlace de datos	✓ Combina bits en bytes y bytes en tramas. Examina y comprueba la dirección y recepción de las tramas. ✓ Acceso a los medios utilizando direcciones MAC. ✓ Comprueba los errores mediante la información del CRC.	802.3, 802.2, HDLC
Física	✓ Mueve bits entre dispositivos. ✓ Especifica el voltaje, la velocidad del cable y la extensión de los cables. ✓ Envía la señal por el medio de transmisión. ✓ Controla los errores de transmisión. ✓ Determina el tipo de señal, si es analógica o digital. ✓ Transforma los datos en una señal de transmisión adecuada para el medio de transferencia.	EIA/TIA-232, 449 V. 24, V.35 G.703 IEEE 802.3 FDDI

Tabla 1.1 Funciones y protocolos empleados en cada capa del modelo OSI

En la figura 1.6 tenemos un ejemplo de cómo un paquete puede ser dividido en capas relacionadas con el modelo OSI.

Capa 1 Física	Capa 2 Enlace de datos		Capa 3 Red		Capa 4 Transporte	Capa 5 Sesión		Capa 6 Presentación	Capa 7 Aplicación
IEEE 802.3	MAC Origen	MAC Destino	IP Destino	IP Origen	No. Protocolo	No. Pto. Origen	No. Pto. Destino	Codificación	TELNET

Figura 1.6 Paquete dividido en las capas del modelo OSI

1.2 Definición de Protocolo

Para que un paquete de datos viaje de un origen a un destino en una red, es importante que todos los dispositivos de dicha red hablen el mismo "lenguaje" o protocolo.

Se darán algunas definiciones de protocolo:

El autor Angel López define el término Protocolo como:

> *"Protocolo: Conjunto de normas que gobiernan la comunicación entre las entidades de red para realizar las funciones encomendadas al nivel correspondiente".*

Por otra parte el autor Michael Palmer nos indica que:

> *"Protocolo: Conjunto de reglas básicas que especifica cómo se formatean los datos de la red en un paquete, cómo se transmite el paquete y cómo se interpreta el paquete en la recepción".*

Ahora bien Cisco Systems nos da su definición:

> *"Un protocolo de red es un conjunto de normas que hacen que sea posible y más eficientemente la comunicación en una red".*

De lo anterior considero que :

> *Protocolo: es un conjunto de reglas y normas, que determinan la manera de cómo los dispositivos de una red van a intercambiar información.*

1.3 Solicitud para comentarios (RFCs)

Las solicitudes RFC (Request For Comment) son los documentos que definen a Internet. Hablan de cómo funciona, cómo usarla y hacía dónde se dirige. La mayoría de las RFC es relativamente técnica. Existen más de 1200 RFCs. Hay un índice en el archivo rfc-index.txt. Algunas RFC están distribuidas en texto, otras en PostScript. Los documentos de texto tienen nombres de la forma *rfcnnnn.txt*, los documentos en PostScript están en archivos con nombres como *rfcnnnn.ps*. En ambos casos, *nnnn* es el número de la RFC que se quiere. Muchas computadoras sólo archivan juegos parciales[6]. Para más información consultar la página http://www.ietf.org/rfc.html.

1.4 Protocolo de Internet (IP)

El Protocolo de Internet (Internet Protocol, IP) actualmente conocido como IP versión 4 (IPv4), es el protocolo encargado del envío y entrega de paquetes. Cada paquete de IP de entrada o salida se denomina *datagrama*. Sin embargo, IP genera datagramas encapsulando la carga con la dirección IP de origen y la dirección IP destino.

Además, IP implementa dos funciones básicas: el encaminamiento y la fragmentación. Para la primera de las funciones se apoya en la dirección IP del host destino. Esta dirección es utilizada para transmitir los datagramas hacia el host correspondiente.

La segunda de las funciones está influenciada por el nivel de enlace. Los datagramas generados por IP deben ajustarse al tamaño máximo que es capaz de transportar dicha red, la cual está limitado por la capa del nivel de enlace. Si el tamaño máximo de una trama es menor que el datagrama generado por IP, entonces la capa IP se ve obligada a fragmentar, de manera que los datagramas resultantes puedan ser enviados por la red.

El protocolo de Internet define una forma para asignar direcciones, utiliza una jerarquía de direcciones de dos niveles, ver figura 1.7. Cada uno de los hosts de internet tiene una dirección única . Para esto se divide una dirección IP en dos partes:

- Dirección de red ó identificador de red (NetId)
- Dirección de host ó identificador de host (HostId)

[6] Krol Ed, Conéctate al mundo de Internet, p. 477

Fig. 1.7 Dirección IP dividido en dos partes, la de red y la de host.

1.4.1 Cabeceras de IP

La cabecera IP se puede dividir en una serie de campos:

- *Versión*: Indica la versión IP. Este campo consta de 4 bits de longitud y contiene la versión 4 ó la versión 6.

- *Longitud de la cabecera*: Indica el número de palabras de 32 bits (4 bytes) de la cabecera IP. Este campo tiene una longitud de 4. Si una opción IP no utiliza 32 bits de una palabra, los restantes se suplen mediante ceros, con el fin de que la longitud de la cabecera sea siempre múltiplo de 32 bits.

- *Tipo de servicio y prioridad*: indica los valores de configuración del tipo de servicio, así como, la calidad de servicio deseada. Este campo consta de 8 bits de longitud y contiene información relativa a la prioridad, retrasos, rendimiento y fiabilidad.

- *Longitud total*: Indica la longitud total de los datagramas IP (la cabecera más la carga). Este campo tiene 16 bits de longitud y contiene una serie de palabras de 32 bits incluidos en el datagrama.

- *Identificación*: Identifica el datagrama IP específico. Este campo tiene 16 bits de longitud. Si el datagrama se fragmenta durante el proceso de enrutamiento, la información de este campo se utiliza para su reensamblado en el destino.

- *Indicador*: contiene los indicadores de fragmentación. Este campo tiene 3 bits de longitud, pero actualmente, se utilizan únicamente dos de ellos. Uno de los bits sirve para indicar si se trata del fragmento final del datagrama (o si va seguido por otros). El segundo se utiliza para señalar si se puede fragmentar el datagrama.

- *Fragmento de desplazamiento*: Indica la posición del fragmento de la carga IP con respecto al original para propósitos de ensamble. Este campo tiene 13 bits de longitud.

- *Tiempo de vida:* Indica el tiempo en segundos que permanecerá un datagrama en la red antes de ser desechado. Cada vez que un datagrama pasa por un

ruteador, se reduce su tiempo de vida en un segundo y se convierte de forma efectiva en un contador de saltos. Este campo consta de 8 bits de longitud.

- *Protocolo*: Indica el protocolo que entregó la carga a IP para su envío. Este campo consta de 8 bits de longitud.

- *Suma de comprobación*: Se utiliza únicamente para comprobar la integridad de la cabecera IP, por lo que suele hacer referencia a este campo como *Suma de comprobación de cabecera*. La carga puede constar de su propia suma de comprobación; la suma de comprobación se vuelve a calcular cada vez que el datagrama pasa por un *ruteador*. Este campo consta de 16 bits de longitud.

- *Dirección de destino*: contiene la dirección IP de destino este campo tiene 32 bits de longitud en IPv4 y 128 en IPv6.

- *Opciones y Rellenos*: especifica las opciones IP. Si existe, su longitud es de 32 bits o de un múltiplo de 32.

En la siguiente figura se muestra un esquema de las cabeceras de IPv4.

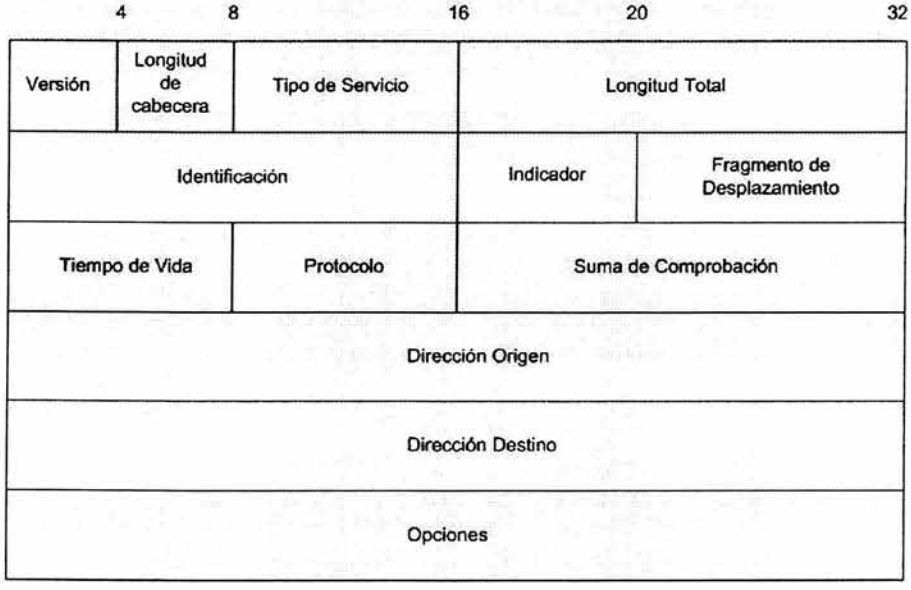


Figura 1.8 Cabecera de IPv4

1.4.2 Direcciones IP

Una dirección IP, también llamada número de IP o dirección de Internet, es una forma de encontrar computadoras en Internet, es un número único, global y estandarizado. Cuando se desea conectar a otra computadora, transferir archivos o enviar un e-mail, primero se debe conocer quien es la otra computadora, es decir su dirección IP. La dirección IP es un "identificador" para que los hosts que desean conectarse a Internet deban estar de acuerdo en usar el mismo esquema de direcciones para establecer una comunicación.

Una dirección IP es un número binario de 32 bits, dividido en 4 campos de 8 bits, separado cada campo por un punto. Las direcciones IP son representadas en notación decimal. Cada byte (octeto) con un valor entre 0 y 255 esta separado por un punto. Cada dirección IP esta asociado a una máscara de 32 bits. La máscara (natural) puede dividir la dirección en dos partes. Una parte identifica la red y la otra parte a la dirección de hosts. Dependiendo del número de bits que destinemos a cada parte, nos podemos encontrar con cuatro tipo de redes.

La asignación de las direcciones IP se cumple a través de una agencia central conocida como Centro de Información de la Red (Network Information Center, NIC), bajo la autoridad del *Internet Assigned Numbers Authorithy* (IANA). El NIC es responsable de asignar direcciones IP únicas para cualquier organización que desee conectarse a Internet. En muchos casos, un Proveedor de Servicios de Internet local (Internet Service Providers, ISP) le solicitará una dirección IP en su nombre o le proporcionará una propia[7].

Las clases de direcciones IP dependen del tamaño de la red (es decir el número de host que la red puede soportar), y deben cumplir con los siguientes requisitos:

- Una dirección de IP es única.
- Dos computadoras no pueden tener asignada el mismo número de IP.
- La dirección IP es global y estandarizada

Así, una forma de distinguir las diferentes clases consiste en usar la regla del primer byte. Con esta regla, el primer byte determina a qué clase pertenece la dirección. Por ejemplo, el uso de la dirección IP 10.1.3.2, 10 es el primer byte de esta dirección. El número 10 cae en el rango de 1-127; por tanto, este dirección IP es clase A y la porción de red es 10; mientras la porción de host es 1.3.2.

Existe un proceso llamado *subneteo* del cual hablaremos más adelante, que reduce el número de direcciones disponibles para los hosts pero incrementa el número de redes disponibles.

[7] Maxwell Steve, Red Hat Linux, p. 87

1.4.2.1 Clasificación de direcciones IP

Existen organizaciones encargadas de repartir las direcciones IP a nivel mundial, por mencionar algunas, ICANN, ARIN, RIPE-NCC, y han asignado dichas direcciones a partir de la división de clases que tiene como arquitectura IPv4 y son de la siguiente manera: Clase A, B, C, D y E. Las direcciones de Clase A para instituciones gubernamentales de todo el mundo, las de clase B para compañías medianas y clase C para los demás solicitantes[8]. En la siguiente figura se muestra un ejemplo de las clases de direcciones IPv4.

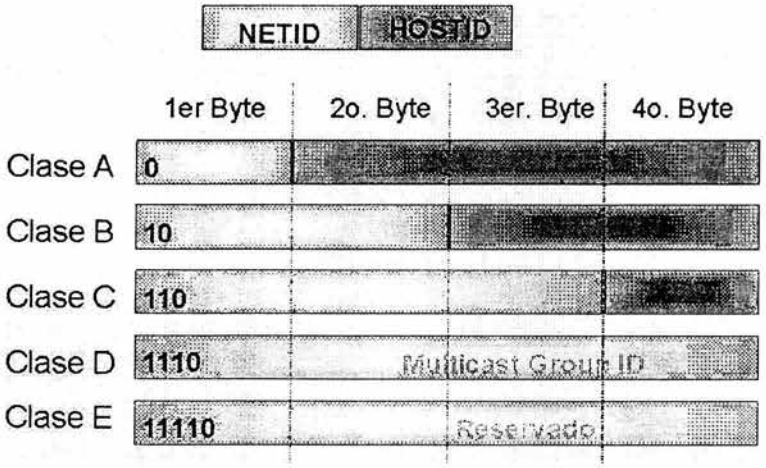


Figura 1.9 Clases de direcciones IP

Cuando un espacio de dirección IP era especificado, estas direcciones se dividía en los grupos o las clases. El formato binario en direcciones define la clase de dirección. En la siguiente tabla se muestra la división por clases de una dirección IP.

Clase	Los bits iniciales	Valor del primer octeto
A	0	1 a 126
B	10	128 a 191
C	110	192 a 223
D	1110	224 a 239
E	11110	240 a 254

Tabla 1.2 Clases de direcciones IP

[8] Guía de segundo año, p. 359

Como se puede observar en la tabla anterior los valores 0, 127 y 255 no están permitidos. La siguiente figura ilustra los rangos correspondientes a cada clase.

Clase A	**0**.0.0.0 *a* **127**.255.255.255
Clase B	**128**.0.0.0 *a* **191**.255.255.255
Clase C	**192**.0.0.0 *a* **223**.255.255.255
Clase D	**224**.0.0.0 a **239**.255.255.255
Clase E	**240**.0.0.0 a **247**.255.255.255

Figura 1.10 Rango de las clases de direcciones

Redes de Clase A

Las direcciones de Clase A se diseñaron para crear redes extremadamente grandes Debido a que las necesidades de este tipo de redes se supuso que serían mínimas, se desarrolló una arquitectura que maximizaba el número posible de direcciones de host, pero limitaba severamente el número potencial de redes de Clase A que se podrían definir.

La identidad de una clase A se define a través del valor del primer octeto (8 bits), como se muestra en la figura 1.11. A causa de que los valores del primer octeto de la dirección de clase A son del 0.0.0.0 a 127.255.255.255, existen 126 redes únicas de clase A. Los 24 bits restantes de la dirección definen al host. El número máximo de hosts de cualquier red de clase A es 2^{24} –2 ó 16,777,214.

El bloque de direcciones de clase A contiene 2^{31} direcciones individuales (incluyendo octetos reservados de valores 0 y 127) y las direcciones IPv4 contienen un máximo de 2^{32} direcciones individuales. La máscara de subred para esta red es la 255.0.0.0

Todas las direcciones IP tienen que ser únicas en su red física. Normalmente una red de clase A 10.0.0.0 *se emplea para asignar direcciones en una intranet,* debido a que esta red, es una red no homologada, y no se puede utilizar de manera global. Si los host en una red 10.0.0.0 acceden a Internet, es necesario utilizar un Traductor de direcciones de red (Network Address Translation, NAT) del cual se hablará más adelante.

Un ejemplo de dirección de red de clase A es el siguiente:

12.120.30.15

donde:

el 12 es la red y 120.30.15 es el host

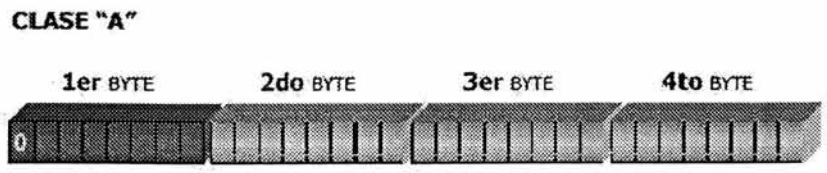


Figura 1.11 Dirección clase A

Redes de clase B

Las direcciones de Clase B se diseñaron para dar respuesta a las necesidades de las redes de tamaño medio a grande. La red se define mediante el valor de los primeros dos octetos o 16 bits, ver figura 1.12. *Los primeros 2 bits* identifican la red como clase B, lo cual proporciona 14 bits para especificar identidades únicas de esta red. Así, 2^{14} ó 16,384 redes de clase B pueden ser definidas, cada una con 2^{16} –2 ó 65,534 hosts. El rango es desde 128.0.0.0 a 191.255.255.255. La máscara de subred de una dirección clase B es 255.255.0.0

Un ejemplo de red de dirección de clase B es el siguiente:

142.16.80.139

donde:

142.16 es la red y el host 80.139

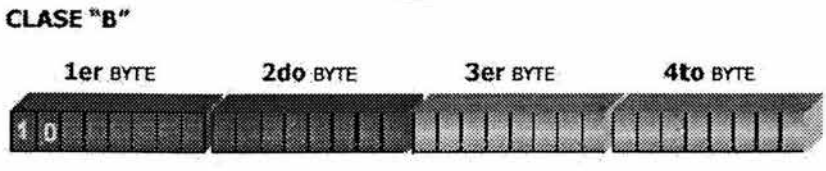


Figura 1.12 Dirección clase B

Redes de clase C

El espacio de direcciones de Clase C es, de lejos, el que más habitualmente se ha utilizado de las clases originales de direcciones IPv4 y se ideó para dar soporte a una red pequeña.

La identidad se define utilizando *los primeros tres valores* del octeto o 24 bits ver figura 1.13. Los primeros 3 bits identifican la red como clase C, los 21 bits restantes se emplean para especificar las identidades únicas de esta red. El número máximo de host de cualquier red de clase C es 2^8 –2 ó 254. El rango de dirección de la red van desde 192.0.0.0 a 223.255.255.255. La máscara de subred de una dirección clase C es 255.255.255.0

Un ejemplo de red de dirección de clase C es el siguiente:

192.100.200.222

donde:

192.100 es la red y 200.222 es el host dentro de ella

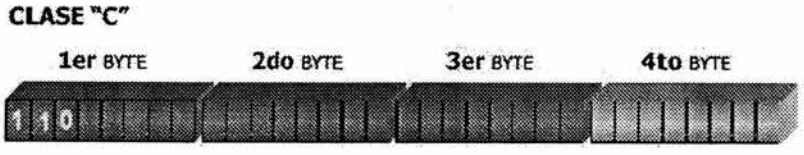


Figura 1.13 Dirección clase C

Redes de clase D y E

Dos clases adicionales (D y E) están definidas, pero están reservadas y no pueden usarse para direcciones de red normales.

Las direcciones de clase D se emplean para un apoyo de multidifusión (*multicast*), como se ve en la figura 1.14. El rango de direcciones de esta clase puede ir desde 224.0.0.0 a 239.255.255.255.

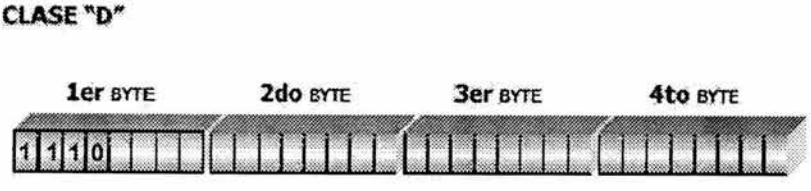


Figura 1.14 Dirección clase D

Las direcciones clase E están reservadas para uso experimental y tienen un intervalo de 240.0.0.0 a 247.255.255.255. Ver figura 1.15

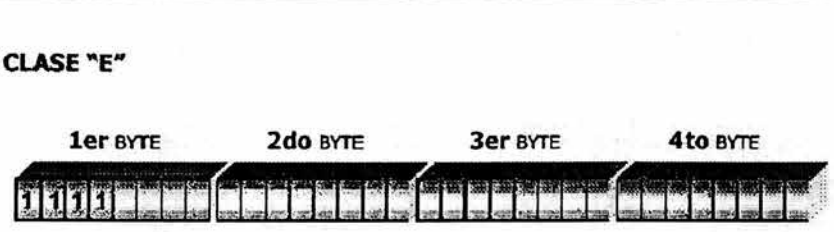


Figura 1.15 Dirección clase E

1.4.2.2 Direcciones especiales

El IP también define algunas direcciones reservadas que incluyen direcciones de ciclo de retorno (*loopback*) y de difusión (*broadcast*). La red *loopback* está definida con la dirección 127.0.0.1 Los datos se envían a los bucles 127.0.0.1 de vuelta al emisor, sin atravesar la red o incluso solo a través de la tarjeta de interfaz de red. Enviando datos de prueba (como un ping) a 127.0.0.1, un host puede probar si su software IP funciona. Para que esta prueba funcione, los tres últimos octetos pueden tener cualquier valor excepto 0.0.0 ó 255.255.255. Todos los hosts utilizan esta dirección para referirse a ellos mismos. Esta dirección de red es reservada y no se debe usar como una dirección de red. De hecho, las especificaciones del protocolo IP no recomiendan su uso en una red activa.

La dirección de difusión (*broadcast*) queda definida como 255 también se considera especial, ya que se usa para direccionar todos los hosts en un rango dado. Por ejemplo, dada una red de 134.110.0.0, la cual es una red clase B, la dirección de difusión de 134.110.255.255 direcciona todos los dispositivos en toda la red 134.110. Debido al significado especial asociado con 255, tampoco debe usarse como dirección de red activa.

En resumen una dirección IP que contiene ceros binarios en todas sus posiciones de host se reservan para la dirección de la red, por ejemplo, 134.110.0.0 es un identificador de red. Las direcciones IP de difusión terminan con unos binarios en toda la parte de host de la dirección, por ejemplo 134.110.255.255.

Es importante el significado de la parte de red de una dirección IP; la cual se llama Identificador de Red (Id. de red) . Los hosts de una red sólo se pueden comunicar directamente con los dispositivos que tengan el mismo ID de red. Pueden compartir el mismo segmento físico, pero si tienen distintos números de red, habitualmente no se pueden comunicar unos con otros, a menos que haya otro dispositivo que puede hacer una conexión entre las redes.

En la tabla 1.3 se muestran las direcciones IP que no son válidas en Internet, comúnmente llamadas direcciones No homologadas:

Direcciones no homologadas	
10.0.0.0	255.255.255
172.16.0.0	172.31.255.255
192.168.0.	192.168.255.255

Tabla 1.3 Direcciones de uso especial que no pueden ser utilizadas o asignadas a una red

1.4.2.3 Subred

Subred es un mecanismo usado para dividir una red en subredes. Su principal objetivo consiste en reducir el tamaño de un dominio de difusión, debido a que, las difusiones se envían a todos los hosts de una red o subred. Cuando el tráfico de difusión empieza a consumir demasiado ancho de banda, los administradores de red pueden reducir el tamaño del dominio de difusión. Para crear una dirección de subred, un administrador toma prestados bits en la parte de host original y los designa como el campo subred.

Las subredes permiten un uso más efectivo de las direcciones existentes. Con las subredes, la porción del nodo de la dirección IP se divide en dos secciones: la dirección de subred y la dirección del host, ver figura 1.16.

Figura 1.16 Subred

Para implementar la subred, se deben cumplir los siguientes requisitos. Primero, se debe crear una máscara de subred para usarla en cada uno de los dispositivos que participarán en la subred. Esta máscara de subred es una dirección especial de 32 bits, la cual se expresa como una dirección IP normal, usando una notación decimal punteada. Como en el caso de la dirección IP, cada uno de los octetos en la subred está en el rango de 1 a 255. Pero a diferencia de las direcciones IP, los octetos representan un grupo de bits enmascarados que se combinan con la dirección IP del dispositivo para darle prioridad a la red de la subred. Determinar la subred involucra combinar la máscara de subred y la dirección IP de host con el operador *Booleano AND*, ver figura 1.17.

Segundo, cada dispositivo que va a participar en una subred debe usar la misma dirección de la máscara de subred. Para cada interfaz definida en el sistema local, la máscara de subred debe ser definida junto con los otros parámetros de la interfaz[9].

De esta manera, para subdividir una red deben de seguirse los siguientes pasos:

1. Determinar la clase de dirección IP de la red que quiere subdividir.
2. Determinar la cantidad de direcciones que se requiere para la subred.
3. Determinar si se puede usar un octeto completo (por lo menos con una dirección clase B) para la dirección de la subred.
4. Realizar *AND booleano* en una de las direcciones IP del dispositivo usando la máscara de subred.
5. Aplicar la máscara de subred a todos los dispositivos que participarán en la subred.

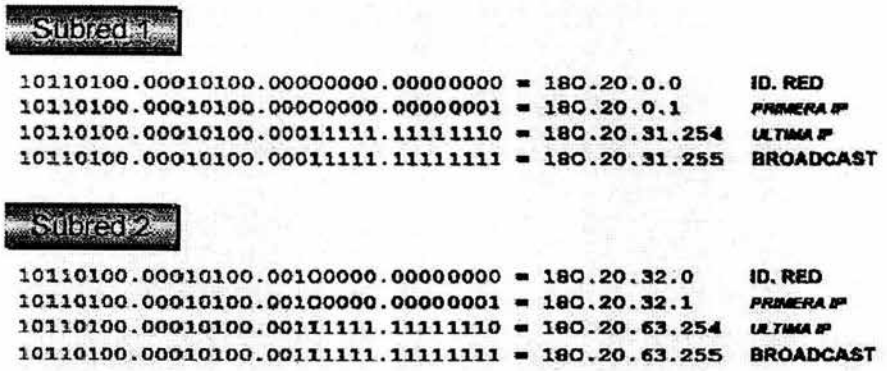


Figura 1.17 Ejemplo de la obtención una subred

1.4.2.4 Máscara de subred

La máscara de subred es un número de 32 bits y cuatro octetos igual que una dirección IP, pero tiene un formato específico. La máscara de subred se escriben normalmente, en formato decimal con puntos (255.255.0.0) o en formato de prefijo (/16), donde los valores después de la barra representan la cantidad de unos. También determina qué parte de una dirección IP es el campo de red y cuál es el campo de host.

[9] Maxwell Steve, Red Hat Linux, p. 88

La función principal de una máscara de subred es identificar qué parte de la dirección IP define la red y qué parte define al host. Los unos (1) especifican que los bits equivalentes de dirección IP son bits de la red y los ceros (0) especifican los bits correspondientes al host. En la clasificación convencional de direcciones IP, la dirección inicial de bits define la clase de dirección, lo cual en sí define el intervalo de valores del host y de la red.

Las máscaras de subred sólo tienen **unos** en su parte de red y de subred, y sólo **ceros** en la parte de host. Por defecto, si no se toman bits prestados, la máscara de subred para una red de Clase B es 255.255.0.0, Una red de Clase C sólo tiene un octeto en el campo host. Por tanto, sólo se pueden tomar prestados hasta 6 bits en estas redes para crear subredes.

La operación binaria *AND* causa los ceros (0) en la máscara de subred para enmascarar la parte del host de dirección IP, dejando sólo los bits que identifican la red. Las direcciones de clase A (/8) tienen por defecto /8 máscaras de subred (255.0.0.0). Las clases B y C tiene máscaras /16 (255.255.0.0) y /24 (255.255.255.0) respectivamente, estás son conocidas como máscaras naturales, ver figura 1.18.

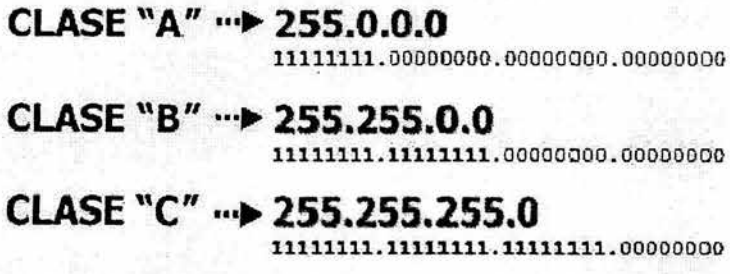


Figura 1.18 Máscaras naturales de las clases A, B y C.

Para determinar la máscara de subred para una dirección IP de subred particular, siga los pasos que a continuación se muestran:

1. Exprese la dirección IP de subred en forma binaria.
2. Sustituya la parte de red y de subred de la dirección por unos.
3. Sustituya la parte de host de la dirección por ceros.
4. Convierta la expresión binaria en notación decimal con puntos.

En la siguiente tabla se muestran las máscaras de subred más utilizadas.

Binario	Decimal	Prefijo
.1000 0000	128	/25
.1100 0000	192	/26
.1110 0000	224	/27
.1111 0000	240	/28
.1111 1000	248	/29
.1111 1100	252	/30
.1111 1110	254	/31

Tabla 1.4 Ejemplo de máscaras de subred de una clase C ó prefijo /24

En la figura 1.19 se muestra un ejemplo de cómo determinar la máscara de subred de una dirección IP dada.

	Red	Subred	Host
Dirección IP del host 132.248.3.130	10000100 11111000	00000011	10000010
Máscara de subred 255.255.255.0	11111111 1111111	11111111	00000000
Subred	10000100 11111000 132 248	00000011 3	00000000 0

Figura 1.19 Ejemplo de una máscara de subred

1.5 Protocolo de Mensajes de Control de Internet (ICMP)

El Protocolo de Internet de Control de Mensajes (Internet Control Message Protocol, ICMP) es usado por los *ruteadores* ó los *gateways* para el intercambio y control de información tal como errores, conectividad entre otros. Ciertamente los mensajes ICMP son generados solamente por *ruteadores* pero también algunos mensajes de ICMP pueden ser generados por hosts.

Los mensajes ICMP pueden ser enviados en varias situaciones:

- Cuando un datagrama no puede alcanzar su destino.
- Cuando un *ruteador* o host no dispone de los recursos necesarios para redirigir un datagrama y por tanto lo descarta. Esta falta de recursos normalmente es producida por una saturación en la red.
- Un *ruteador* no puede redirigir un datagrama a otra red porque es demasiado grande y sería necesario fragmentarlo, pero en la cabecera IP del datagrama no se permite la fragmentación.
- Cuando un host o *ruteador* puede informar al host origen sobre un camino más corto para que el datagrama alcance su destino.

Los mensajes ICMP no son generados en los siguientes casos:

- Cuando un mensaje de error ICMP es recibido.
- Cuando una dirección de datagrama de loopback, broadcast o multicast es recibido.
- Cuando un fragmento IP no iniciado es recibido.

Existen algunas reglas que hay que seguir a la hora de generar mensajes ICMP:

- No se generará un mensaje de error ICMP para un datagrama que esté transportando un mensaje ICMP.
- No se generará mensajes ICMP para los datagramas que especifiquen un dirección de tipo broadcast.
- Cuando un datagrama provoca el envío de un mensaje ICMP, y que este datagrama ha sido fragmentado, sólo se genera el mensaje ICMP una vez evitando que se envíe un mensaje de error por cada fragmento del datagrama.

Dos clases de mensajes ICMP son definidas: mensajes de error ICMP y mensajes de petición. En general un *ruteador* envía mensajes ICMP para reportar un error. Si un *ruteador* descarta un datagrama por falta de memoria, esto manda un mensaje llamado *source quench* o por falta de recursos que origino el datagrama. Un *ruteador* envía mensajes ICMP llamado *destination unreachable* cuando recibe paquetes de datos para un destino que no conoce o que no sabe como buscar. Un *ruteador* envía mensajes nombrado *redirect* para aconsejar que un *ruteador* diferente es más apropiado para enviar los paquetes a un destino en particular. Los mensajes de error de ICMP incluyen la cabecera del datagrama IP cual es la causa del error.

Generalmente se envían mensajes ICMP para probar la conectividad de la red. Para ello usa el mensaje ICMP llamado *echo*. Un host que recibe un mensaje de *echo* responde con un mensaje ICMP llamado *echo reply*. Estos mensajes son usados para probar la conectividad de la red y también estima un tiempo de **un viaje redondo**.

Los mensajes ICMP viajan por la red dentro del campo de datos de los datagramas IP. En la cabecera IP el campo protocolo se establece a 1 para indicar

que viaja un mensaje ICMP. Todos los mensajes ICMP comienzan con tres campos fijos:

- *Tipo*: Indicando el tipo de mensaje ICMP
- *Código*: ofrece más información sobre el tipo de mensaje, ya que dentro de un tipo podemos tener varios subtipos de mensajes.
- *Checksum*: Es una suma de control, para comprobar la integridad del mensaje.

1.5.1 Tipos de mensajes ICMP

Destino inaccesible (Destination Unreachable)

Este mensaje puede ser enviado en las siguientes situaciones:

El destinatario no puede alcanzarse. Esta situación podría ser detectada por un *ruteador* al comprobar que la dirección destino del datagrama está marcada como inalcanzable en sus tablas de encaminamiento. Es obligación del *ruteador* avisar al host origen del datagrama de esta situación a través de este mensaje ICMP.

El tipo será ICMP_DEST_UNREACH.

Fin de tiempo de espera (Time Exceeded)

Si a un host le va llegando un datagrama fragmentado y no puede completarla reconstrucción del datagrama, debido a la falta de algún fragmento, y no ha llegado en el tiempo de espera que tenga establecido, entonces debe descartar lo que le haya llegado del datagrama hasta ese momento y avisar al host origen de esta situación.

El campo tipo tendrá el valor ICMP_TIME_EXCEEDED.

Problema de parámetro (Parameter Problem)

Se envía un mensaje ICMP de este tipo cuando el host destino del datagrama o un *ruteador* encuentre algún problema con los parámetros de la cabecera del mismo (valores incorrectos en alguna opción), de tal forma que no se pueda procesar el datagrama. En este caso, el datagrama se descarta avisando al host origen de esta situación.

El campo tipo tomará el valor ICMP_PARAMETERPROB.

Paquetes de control de flujo (Source Quench)

Situaciones en las que un mensaje ICMP de tipo *Source Quench* podrá ser enviado:

- Un *ruteador* descartará un datagrama si no dispone de los *buffers* necesarios para almacenarlo y reenviarlo a la red adecuada para que alcance el host destino. Si un *ruteador* descarta un datagrama por este motivo, deberá avisar al host origen de este suceso.
- El host destino podría enviar este mensaje si le están llegando los datagramas demasiado rápido y no le da tiempo a procesarlos. En este caso, el host origen debe interpretar el mensaje como una petición para que el envío de los datagramas se haga más despacio, ya que el host destino no puede procesarlos a la velocidad actual.
- El host origen, ante la llegada de un mensaje de tipo *Source Quench*, deberá disminuir la velocidad a la que envía los datagramas hasta que deje de recibir avisos del *ruteador*. A partir de entonces podría intentar aumentar la velocidad hasta que reciba de nuevo un aviso *Source Quench*.

El campo tipo tendrá el valor ICMP_SOURCE_QUENCH.

Redirección (Redirect Message)

Supongamos la siguiente situación: Un *ruteador* R1 recibe un datagrama de un host A que se desea comunicar con un host B. El host A se encuentra en una red a la cual el *ruteador* R1 también está conectado. Al recibir el datagrama, el *ruteador* R1 comprueba sus tablas de encaminamiento para saber a cuál de las redes a las que está conectado debe redirigir el datagrama. En este caso supondremos que el camino más corto para que el datagrama llegue a su destino es a través del *ruteador* R2. Si se da el caso de que el *ruteador* R2 se encuentra en la misma red que el host origen, entonces el *ruteador* R1 enviará un mensaje de ICMP de tipo *redirect* al host origen, indicándole la dirección IP del *ruteador* óptimo a través de cual debe enviar los datos para llegar al host destino. A partir de este momento el host A tendrá que actualizar sus tablas de encaminamiento para que en futuros envíos sea el *ruteador* R2 el encargado de encaminar los datagramas hacia el host B. No obstante, en esta primera ocasión será el *ruteador* R1 quien, tras realizar el aviso, reenvía el datagrama hacia su destino, que es el host B. Ver figura 1.20

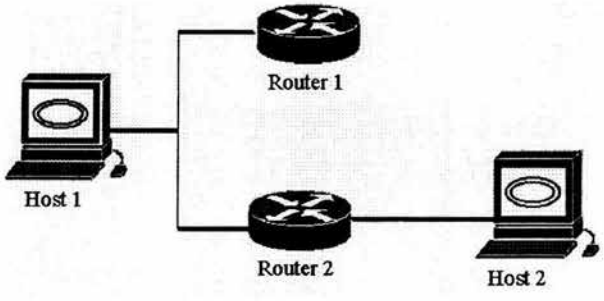


Figura 1.20 Mensaje redirect

Si se está realizando encaminamiento en origen, el *ruteador* no enviaría el aviso, sino que reenviaría el datagrama aunque el camino seguido no sea el más óptimo.

El campo tipo tomará el valor ICMP_REDIRECT.

Eco y respuesta de eco (Echo y Echo Reply)

Los datos recibidos en un mensaje *echo* serán devueltos en un mensaje de tipo *echo reply*, como se muestra en la figura 1.21.

Los campos identificador y número de secuencia serán útiles desde el punto de vista del proceso que envía los mensajes *echo*, ya que le ayudarán a diferenciar los mensajes ICMP que correspondan a respuestas a sus mensajes echo de aquellos que no lo sean.

El campo Tipo podrá ser ICMP_ECHO ó ICMP_ECHOREPLY.

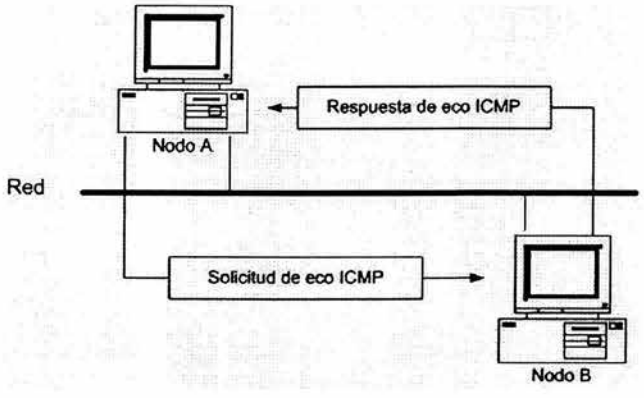


Figura 1.21 Mensajes de echo

Marca de tiempo (Timestamp y Timestamp Reply)

Los datos recibidos en un mensaje de tipo *Timestamp,* consistentes en una marca de tiempo, serán devueltos junto a una marca de tiempo adicional.

- La marca de tiempo de envío identifica el momento, inmediatamente anterior, al envío del mensaje de tipo *timestamp* por el proceso emisor.
- La marca de tiempo de recepción identifica el momento en el que el proceso receptor lee el mensaje.
- La marca de tiempo de transmisión identifica el momento en que el proceso receptor envía de vuelta el mensaje.

El campo tipo podrá tomar los valores ICMP_TIMESTAMP ò ICMP_TIMESTAMPREPLY.

Solicitud/Respuesta de información (Information Request e Informtation Reply)

Este tipo de mensaje puede servirle a un host para obtener el número de la red en la que se encuentra. En la cabecera IP los campos de dirección origen y dirección destino tendrían los bytes que identifican a la red con valor cero. La capa IP del host destino rellenaría estos campos correctamente, de tal manera que el host origen averiguaría a qué red se encuentra conectado al recibir la respuesta.

El campo tipo podrá tomar los valores ICMP_INFO_REQUEST ó ICMP_INFO_REPLY.

Dirección de máscara de subred (Address mask)

En este mensaje permite a un host conocer la máscara de subred con la que debe trabajar. Para ello el host envía un mensaje de tipo *Address Mask Request* a toda la red (broadcast). Si en la red hay un servidor autorizado que pueda ofrecer esta información, se envía como respuesta al host la máscara de subred que debe usar, poniendo a unos en el campo Máscara, la parte correspondiente a red y subred.

El campo tipo podrá ser ICMP_ADDRESS ó ICMP_ADDRESSREPLY.

A continuación se dan algunos ejemplos de mensajes ICMP:

Hydra> ping 192.100.199.134
PING 192.100.199.134: 56 data bytes
64 bytes from 192.100.199.134: icmp_seq=0. time=25. ms
64 bytes from 192.100.199.134: icmp_seq=1. time=34. ms
64 bytes from 192.100.199.134: icmp_seq=2. time=33. ms
64 bytes from 192.100.199.134: icmp_seq=3. time=30. ms
64 bytes from 192.100.199.134: icmp_seq=4. time=8. ms
64 bytes from 192.100.199.134: icmp_seq=5. time=99. Ms
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms) min/avg/max = 8/32/99

Hydra> ping 132.248.3.1
PING 132.248.3.1: 56 data bytes
ICMP Port Unreachable from gateway bufadora.astrosen.unam.mx (132.248.3.1)
for icmp from hydra (200.15.3.86) to bufadora.astrosen.unam.mx
(132.248.3.1)
ICMP Port Unreachable from gateway bufadora.astrosen.unam.mx (132.248.3.1)

Hydra> ping 192.168.1.1
PING 192.168.1.1: 56 data bytes

----192.168.1.1 PING Statistics----
4 packets transmitted, 0 packets received, 100% packet loss

Hydra> ping 192.100.199.134
ICMP Time exceeded in transit from 192.100.199.134
for icmp from hydra (132.248.120.53) to quetzal.innsz.mx (200.15.34.61)
ICMP Time exceeded in transit from 192.100.199.134
for icmp from hydra (132.248.120.53) to quetzal.innsz.mx (200.15.34.61)
ICMP Time exceeded in transit from 192.100.199.134

Hydra> ping 200.15.34.61
PING 200.15.34.61: 56 data bytes
64 bytes from quetzal.innsz.mx (200.15.34.61): icmp_seq=0. time=11. ms
64 bytes from quetzal.innsz.mx (200.15.34.61): icmp_seq=1. time=10. ms
64 bytes from quetzal.innsz.mx (200.15.34.61): icmp_seq=2. time=18. ms
64 bytes from quetzal.innsz.mx (200.15.34.61): icmp_seq=3. time=155.ms
64 bytes from quetzal.innsz.mx (200.15.34.61): icmp_seq=4. time=158. ms
64 bytes from quetzal.innsz.mx (200.15.34.61): icmp_seq=5. time=95. Ms

----200.15.34.61 PING Statistics----
5 packets transmitted, 5 packets received, 2% packet loss
round-trip (ms) min/avg/max = 9/21/306

1.6 Protocolo de Resolución de Direcciones (ARP) y Protocolo de Resolución Inversa de direcciones (RARP)

Para que dos computadoras de una determinada red se puedan comunicar, cada una debe de conocer la dirección física (Medium Access Control, MAC) de la otra. Por medio de la difusión del ARP, un host puede de manera dinámica, descubrir la dirección de la capa MAC correspondiente a una dirección IP. Por lo tanto ARP se utiliza para resolver ó *asignar una dirección IP conocida a una dirección de subcapa MAC.* Para determinar la dirección de destino para un datagrama, se consulta la tabla de memoria caché ARP. Si la dirección no figura en la tabla, ARP envía un *broadcast* que busca la estación destino. Cada estación en la red recibe el *broadcast* y solo la máquina que la tiene contesta enviando su dirección MAC que es colocada en la tabla de ARP del equipo origen.

RARP se utiliza para mapear direcciones de la capa MAC con direcciones IP, RARP que es la lógica inversa de ARP, puede ser utilizado por estaciones de trabajo sin disco que no conozcan sus direcciones IP cuando se inicializan, RARP se basa en la presencia de un servidor RARP que cuente con una entrada en la tabla, o en otro medio para responder a las solicitudes RARP. En el segmento local, se puede utilizar RARP para iniciar una secuencia de carga de sistema operativo a distancia.

La operación básica de ARP es simple: cuando el nodo A quiere determinar la dirección física del nodo D, el nodo A enviará un mensaje de difusión solicitando la

dirección del nodo D a todos los dispositivos en la red. Como se muestra en la figura 1.22 todos los hosts reciben el mensaje, incluidos el nodo B y el nodo C, pero sólo el nodo D responderá con la dirección física correcta, como lo muestra la figura 1.23. La razón por la que sólo el nodo D responde es que éste examina el paquete ARP y determina su propia dirección IP coincide con la dirección IP objetivo que está buscando el solicitante.

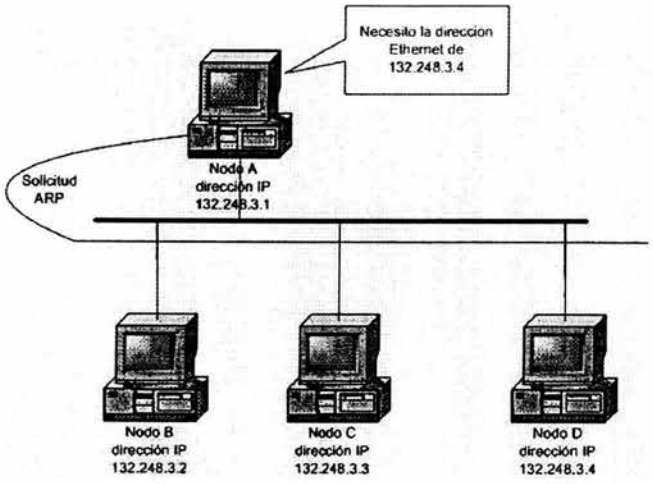


Figura 1.22 Solicitud ARP

RARP hace la inversión; dada una dirección MAC, encuentra la dirección IP correspondiente.

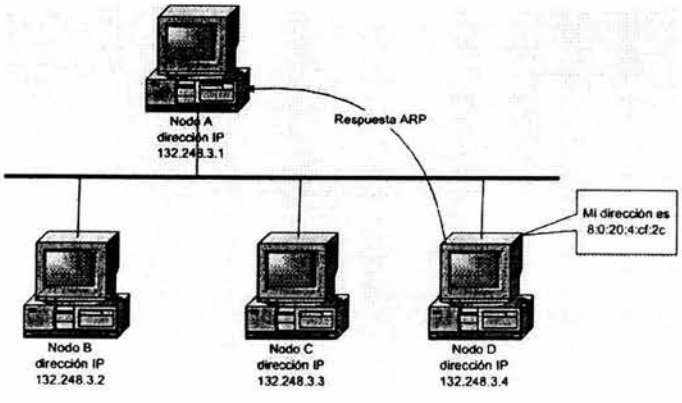


Figura 1.23 Respuesta ARP

Las direcciones MAC se expresan en 48 bits y están separadas por dos puntos (:) Esta notación de dos puntos se usa como el primer método para representar estas direcciones de hardware. La autoridad del registro de el Instituto de Ingenieros en Electrónica y Electricidad (Institute of Electrical and Electronics Enginers, IEEE) le asigna la porción del vendedor a aquellas organizaciones que producen hardware de trabajo en red y solicitan un código de vendedor. Estos códigos también son conocidos como identificadores únicos de organización (Organization Unique Identifiers, OUI). Los tres primeros bytes de la dirección representan el número de serie de la unidad[10], como se muestra en la siguiente figura.

Fig. 1.24 Formato de la dirección de enlace de datos

1.7 Sistema de Nombres de Dominio (DNS)

El Sistema de Nombres de Dominio (Domain Name System, DNS) es un servicio de resolución de nombres que resuelve (asocia) nombre de host a direcciones IP. El DNS mantiene un registro de direcciones IP y nombres de hosts en un proceso llamado dominio. DNS proporciona servicios a lo largo de una cadena jerárquica, con un diseño de base de datos similar a una estructura de árbol de archivos (nivel raíz/nivel superior/segundo nivel/nombre de host). DNS también presta servicios a las peticiones de nombres de host que no puedan ser resueltas a nivel local[11].

El *Sistema de Nombres de Dominio* **es esencialmente una base de datos de información de equipos**.

La organización responsable del manejo del sistema de nombres de dominio es la ICANN (de sus siglas en inglés, Internet Corporation for Assigned Names and Numbers). Dada la expansión internacional del Internet, la ICANN decidió reservar dominios para cada país. Así *mx* para México y *br* para Brasil y así sucesivamente, como se muestra en la figura 1.25

[10] Maxwell Steve, Red Hat Linux, p. 104
[11] Guía del segundo año, Cisco Systems, p 465

La mayoría de los países adoptaron una estructura análoga de dominios para las diferentes afiliaciones de las organizaciones. De tal manera que se pueden encontrar en nuestro país dominios como:

com.mx	Organizaciones comerciales
edu.mx	Organizaciones educacionales
gob.mx	Organizaciones gubernamentales
net.mx	Organizaciones de infraestructura de red.

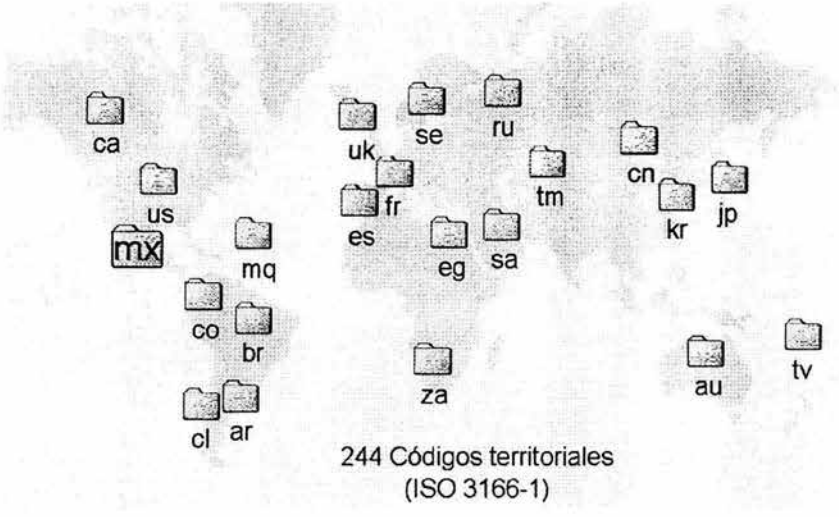


Figura 1.25 Códigos territoriales

Las organizaciones encargadas de la asignación y reasignación de las direcciones IPv4 e IPv6 son actualmente: **ARIN** (American Registry for Internet Numbers), RIPE-NCC (Réseaux IP Européens-Network Coordination Centre), APNIC (Asia Pacific Network Information Centre) y LACNIC (Latin-American and Caribbean IP Address Registry) AFRINIC, como se muestra la figura 1.26, cada una tiene su región de asignación; así que, por ejemplo, para la **UNAM** tiene delegado dos clase B y una clase C de direcciones IPv4 que fueron asignados por **ARIN**, aunque debió de ser LACNIC la organización quién delegará esas clases de direcciones pero cómo todavía no existía se obtuvieron por medio de **ARIN** que era la organización que existía en ese tiempo y por lo tanto le tocaba designar en la región de América.

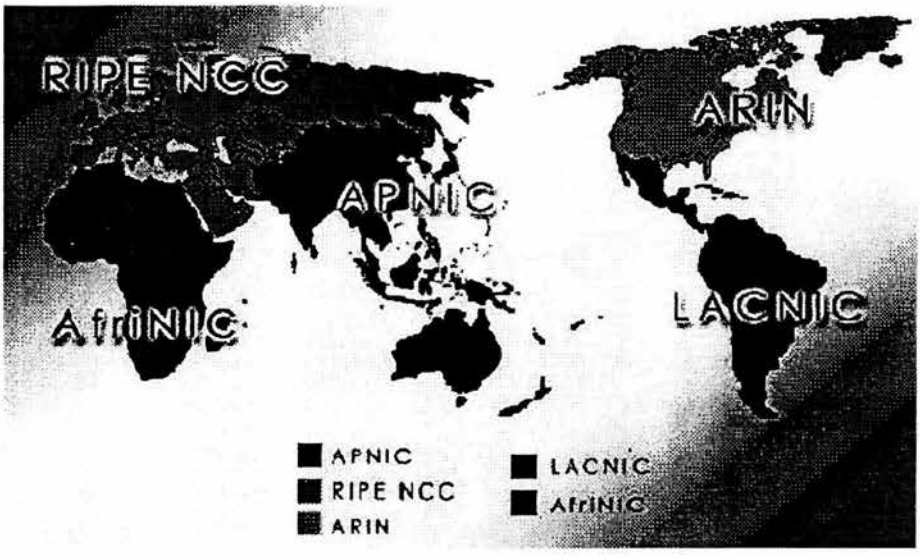


Figura 1.26 Principales organismos de distribución de direcciones IP

1.8 Seguridad

Sabemos hoy en día que la seguridad es uno de los puntos débiles de internet. La seguridad precisa una infraestructura adicional. La seguridad en internet se ve amenazada, entre otras cosas, por:

- Intrusiones físicas o lógicas en uno o más de los elementos de la comunicación.
- Enmascaramiento *"spoofing"* de la identidad que permite accesos no autorizados, ciber-delincuencia, fraudes en la facturación, ataques con virus.
- Violación de la confidencialidad de las comunicaciones, datos de los usuarios métodos de pago...
- Denegación de servicio, repudio, manipulación de información por agentes de interdemediación,...
- Violación a los Derechos de Propiedad Intelectual (música, videos, ...)

La consecución de la Seguridad precisa medidas de 4 tipos:

- Legales
- Técnicas
- Económicas
- De concienciación y formación

1.8.1 IPsecurity (IPsec)

IPsec está diseñado para proporcionar un nivel de seguridad de conexión de red (en la capa IP) de alta calidad e interactivo para IPv4 y también IPv6. IPsec logra este proceso al ofrecer "servicios" de seguridad como, por ejemplo, control de acceso, integridad fuera de línea, autenticación del origen de la información, protección contra reconexiones y confidencialidad. Todos estos servicios se proporcionan mediante tres principales protocolos de seguridad: Encabezado de autenticación (AH), carga de seguridad de encapsulación (ESP) y la administración de claves (Intercambio de claves Internet, IKE). IPsec no establece cuál de los protocolos se debe usar; le permite a los sistemas que se comunican negociar el protocolo que deseen usar. Esto se aplica tanto para el protocolo como para los algoritmos, lo cual también significa que no se establece el algoritmo que se debe usar en cada uno de los protocolos; por lo tanto puede usar su propio algoritmo, siempre y cuando la parte con la que se comunique soporte el mismo algoritmo.

Los componentes fundamentales de la arquitectura de la seguridad IPsec son las siguientes:

- *Asociaciones de seguridad.* La forma en que los sistemas determinarán los protocolos y algoritmos que se deben usar. Lo que son, cómo funcionan y la forma en que se administran
- *Protocolos de seguridad.* Encabezado de autenticación (AH) y Carga de seguridad de encapsulación (ESP)
- *Administración de claves.* Manual y automática (el intercambio de clave de Internet [IKE])
- *Algoritmos.* Para la autenticación y cifrado.

Asociaciones de seguridad (SA)

Una Asociación de seguridad es un conjunto de directivas y claves que se usan para proteger la información de una red. Las asociaciones de seguridad son muy importantes para IPsec debido a que se usan para definir los servicios de seguridad, mecanismos y las claves que se usan para proteger la comunicación de un extremo a otro. AH y ESP hacen uso de las SA para obtener las claves que necesitan para realizar su función. Además, uno de los trabajos que hace IKE es establecer y mantener las SA. Una SA está formada por una combinación única de un índice de parámetros de seguridad SPI), una dirección de destino IP y un identificador de protocolo de seguridad (AH o ESP)[12]. La dirección destino puede ser una de las siguientes:

- Dirección única de transmisión
- Dirección de difusión IP

[12] Cox Philip, WINDOWS 2000 Manual de seguridad, p. 484, 485

- Direcciones de grupo multidifusión

Antes de que sea posible el cambio de información de seguridad, se debe establecer una SA entre los sistemas que se comunican. Los sistemas deben acordar la forma en que intercambiarán y protegerán la información. Existen dos tipos de SAs definidas: modo de transporte y modo de túnel. Una SA en modo de transporte es una asociación de seguridad entre dos equipos anfitrión. El modo de túnel se usa cuando ambos extremos son una puerta de entrada de seguridad.

Administración de claves SAs

Todas las implementaciones IPsec deben soportar la administración de SAs y claves, manual y automática. A pesar de que AH y ESP son bastante independientes de las técnicas pueden afectar parte de la funcionalidad que cada uno de estos elementos proporciona (es decir, la distribución detallada de claves determina la autenticación detallada que se proporciona).

- *Manual.* Un individuo configura de forma manual cada sistema con el material de claves y la información de la administración de la asociación de seguridad. Esto sólo funciona para una cantidad limitada de conexiones.

- *Automatizada.* La administración de SA y el material de claves es automático. Esta es la única técnica posible para una implementación de tamaño grande. El protocolo de administración automático predeterminado que se seleccionó para el uso de IPsec es Intercambio de Claves de Internet (IKE).

Protocolos de seguridad

Se pueden proporcionar servicios de seguridad entre un par de equipos que se comunican, un par de puertas de entrada de seguridad que se comunican o entre una puerta de entrada de seguridad y un equipo. IPsec tiene tres protocolos particulares que se usan en diferentes configuraciones para proporcionar los servicios de seguridad.

Encabezado de autenticación: sólo autenticación. AH se usa para proporcionar los servicios de integridad del mensaje, autenticación y la protección opcional contra reconexiones o datagramas. IP. Realiza todas estas tareas sin integrar el concepto de confidencialidad (cifrado). Sin embargo, se puede usar en conjunto con ESP para proporcionar confidencialidad. Todas estas opciones se determinan durante la negociación SA.

Carga de seguridad de encapsulación: autenticación y cifrado. ESP se puede usar para proporcionar los mismos servicios de seguridad que ofrece AH, pero también proporciona un servicio de confidencialidad (cifrado). Las principales diferencias entre la autenticación que se proporciona mediante ESP y AH son que ESP cifra la información y no toma en cuenta la información del encabezado (es decir, protege la información y no los campos del encabezado IP). En realidad, se basa en la

fuerza mínima de cifrado DES de 56 bits, pero se puede usar con casi cualquier algoritmo de cifrado simétrico. Debido a que ESP se usa cifrar la información al desfigurarla y después la acomodarla en otro paquete IP, se puede usar en redes que hoy en día se consideran antiguas.

Funciones de intercambio de claves: intercambio de claves internet. Debido a que AH y ESP usan información de secretos compartidos para proporcionar sus servicios, debe existir una forma para negociar claves; en este punto es cuando entra en acción IKE. IKE es el protocolo de intercambio de claves y negociación de seguridad predeterminado que se usa en las implementaciones IPsec. Su propósito es establecer asociaciones de seguridad y claves de cifrado. En la parte SA del proceso negociará, establecerá, modificará y eliminará SAs y sus atributos. Para la administración de claves, IKE soporta la distribución tanto automática como manual de las claves públicas (asimétricas) y privadas (simétricas).

Modos de asociación de seguridad

Como mencionamos antes, AH y ESP proporcionan soporte para dos modos de uso: modo de transporte y modo de túnel. IPsec permite un control muy específico de las conexiones que se pueden permitir. Puede tener una configuración de equipo anfitrión, equipo anfitrión a puerta de entrada de seguridad y puerta de entrada de seguridad a puerta de entrada de seguridad. El administrador controla todas estas configuraciones. Los dos modos disponibles se describen a continuación:

Transporte (host a host). El modo de transporte protegerá la carga del paquete IP, pero no al encabezado. Piense en este modo como en uno de los protocolos de protección de las capas más altas. Hay que recordar que sólo protege la carga IP pero no al encabezado.

De túnel (puerta de entrada de seguridad a puerta de entrada de seguridad y equipo a puerta de entrada de seguridad). Este modo realiza un sistema túnel en la capa tres del modelo OSI, la carga que viaja a través del túnel es un paquete de capa de red. El paquete IP completo se asegura y protege durante su transferencia mediante uno de los protocolos de seguridad IPsec. En este caso, protege al paquete completo.

Modo túnel ESP. El encabezado IP original (que es el encabezado del paquete original), normalmente incluye las direcciones fuente y destino reales, mientras que el encabezado IP externo contiene las direcciones de una puerta de entrada de seguridad. El encabezado original se coloca después del encabezado ESP, y se agrega una secuela ESP antes del cifrado. Todo lo que aparece después del encabezado ESP se cifra, excepto por la secuela ESP. Ahora el paquete original completo está protegido (cifrado). La información cifrada proviene de la información de un paquete nuevo (es decir, encapsulado), y el encabezado IP

nuevo se usa para rutear el paquete de su origen al siguiente destino, que es normalmente una puerta de entrada de seguridad.

Modo de túnel AH. Es muy similar al modo túnel ESP, pero no se realiza el cifrado del paquete, sólo la autenticación e integridad de la información. Todo el paquete se prepara para asegurar la integridad, incluyendo el nuevo encabezado de túnel.

IPsec y la traducción de direcciones de red (NAT)

La tarea de la traducción de direcciones de red (Network Address Traslation, NAT) es revisar los paquetes y traducir la información de las direcciones que se usan, mientras que el trabajo de IPsec es lograr que los paquetes sean ininteligibles y no se puedan modificar, por lo tanto tenemos una incompatibilidad fundamental. Esta incompatibilidad fundamental significa que no puede establecer la seguridad IPsec de extremo a extremo a través de las puertas de entrada NAT. La puerta de entrada NAT se puede usar como una puerta de entrada de seguridad, y luego reenviar las conexiones IPsec a los hosts internos. No puede crear un túnel para que se realice la conexión debido a que NAT tiene que ver la información de la dirección del datagrama original (es decir, descubrir las envolturas que aplicó IPsec) pero más tarde puede proteger de nuevo los paquetes antes de permitirles la entrada o enviarlos al exterior.

IPsec no es una solución para todo; requiere de una buena implementación y una arquitectura de seguridad bien detallada.

Cifrado

El cifrado de la información de la autenticación no representa un concepto muy difícil, pero es importante asegurarse que la solución que escoja no ponga en riesgo los mecanismos de autenticación, un ejemplo de los riesgos consiste en enviar la información de autenticación que se puede volver a usar (es decir, contraseñas) sobre la red en forma de texto sencillo.

En cuanto al cifrado de información deberá determinar si es que lo necesita o no. Si no necesita el cifrado de la información, entonces puede tomar en cuenta las soluciones de acceso remoto; cuando determine que el cifrado es necesario, deberá determinar la necesidad de su "robustez".

Fortalecer un sistema es una medida que depende de distintos factores. Tiene control sobre dos de estos factores: algoritmo de cifrado y longitud de la clave.

La longitud de la clave es una cuestión mucho más simple: de acuerdo al algoritmo mismo, una clave más larga es una clave más robusta. Aunque tenga en cuenta que usará estas claves para realizar operaciones de computadoras, de forma que una clave larga es también una solución, desde el punto de vista de la computación, mas costosa ; por lo tanto podría afectar al rendimiento de los sistemas de forma significativa.

Secure Shell (SSH)

El protocolo de cuenta segura (SSH) es otro protocolo que se usa con frecuencia para establecer conexiones seguras entre un cliente y un servidor. Principalmente se usa para proteger el inicio de sesiones remotas y la copia segura sobre una red insegura. También puede realizar "capacidades de túnel" del tráfico basado en la sesión protegida sobre su red, de forma que también se puede proporcionar un servicio de red seguro. SSH es un protocolo de la capa de transporte que se ejecuta sobre TCP/IP[13]. Proporciona la funcionalidad siguiente:

- Túnel (también conocido como reenvío de puertos), el cual proporciona servicios seguros de red
- Cifrado robusto
- Autenticación de servidor
- Autenticación de mensajes
- Compresión de paquetes
- Intercambio de claves
- Soporte de claves públicas y simétricas

1.9 Traducción de Direcciones de Red (NAT)

Para que los paquete fluyan en Internet, las direcciones deben de ser direcciones IP "públicas", homologadas o validas (como se les quiera llamar). Como ya se ha comentado a lo largo de este capitulo, hasta hace unos años no representaba ningún problema pero debido a que el número de direcciones era muy grande, pero con el advenimiento de la Web, las direcciones disponibles ahora son escasas. Lo anterior en conjunto con el hecho de que todos los proveedores de servicios de internet (ISP) intentan abarcar tanto espacio de las direcciones IP como les sea posible para venderlas a sus clientes, esto conlleva a la falta de direcciones.

Es debido a esta escasez se han definido las direcciones "privadas". Las direcciones privadas son parte de las direcciones públicas a las cuales se asignan características especiales.

Recordemos cuales son las direcciones privadas:

10.0.0.0 – 10.255.255.255 con máscara de subred 255.0.0.0
172.16.0.0 – 172.31.255.255 con máscara de subred 255.240.0.0
192.168.0.0 – 192.168.255.255 con máscara de subred 255.255.0.0

Debido a que las direcciones "privadas" no pueden recibir tráfico de Internet (es decir, no se pueden rutear a través de Internet), debe existir una forma para

[13] Cox Philip, WINDOWS 2000 Manual de seguridad, p. 490

traducir estas direcciones a direcciones públicas. En otras palabras, un método de traducción de direcciones de red!!

Un ruteador que implementa NAT, puede traducir direcciones "privadas" a direcciones "públicas" (es decir, a direcciones que se pueden rutear en Internet). Al realizar esta traducción, sólo las direcciones internas que se definen en la tabla de asignaciones NAT se podrán acceder por los hosts externos. Por lo mismo, NAT proporciona protección para los ataques basados en Internet, debido a que los paquetes que tienen direcciones privadas como dirección de destino no se rutearán a través de Internet. Por lo tanto, para que un agresor logre que su ataque funcione debe conocer las asignaciones actuales de las direcciones IP internas (privadas) para las direcciones IP externas (públicas) que el ruteador NAT ha definido en sus tablas[14].

Existen dos métodos básicos de NAT:

- ✓ **Muchas a muchas**
- ✓ **Muchas a una** (mejor conocido como mascarada IP)

Se pondrá dos escenarios para ejemplificar lo anterior.

Muchas a muchas

Supongamos que una empresa compra 500 equipos y dicha empresa tiene asignado una red clase C, que como bien sabemos tiene 256 direcciones de las cuales 254 se pueden usar. Para darle salida a estos equipos se puede configurar NAT en un ruteador, por ejemplo, para que soporte *muchas a muchas* con una dirección 10.0.0.0 para el rango de la red privada Esto permitirá que hasta 254 hosts accedan a Internet al mismo tiempo. Cada vez que una computadora necesite acceder a Internet, el ruteador que tiene NAT configurado, asignará una de las direcciones públicas (que se obtiene del espacio de dirección clase "C") a una dirección interna, y luego reenviará los paquetes de un lado a otro.

Muchas a una (mascarada IP)

Este escenario es el más usado, por ejemplo, se configura un ruteador (para nuestro caso) con una dirección pública y cada solicitud de algún host interno que llegue al ruteador saldrá con la dirección pública. Por lo tanto cada computadora con una dirección privada podrá comunicarse a Internet por la dirección publica que se tiene configurada en el equipo (ruteador).

Para realizar NAT no es necesario un ruteador, también existen otros equipos que pueden fungir como ruteadores, tal es el caso de las PC´s, claro con un software adicional.

[14] Cox Philip, WINDOWS 2000 Manual de seguridad, p. 412, 413

A continuación se ilustrará dos casos típicos de NAT, la figura 1.27 ejemplifica el caso en el cual se utiliza un ruteador que tiene configurado NAT.

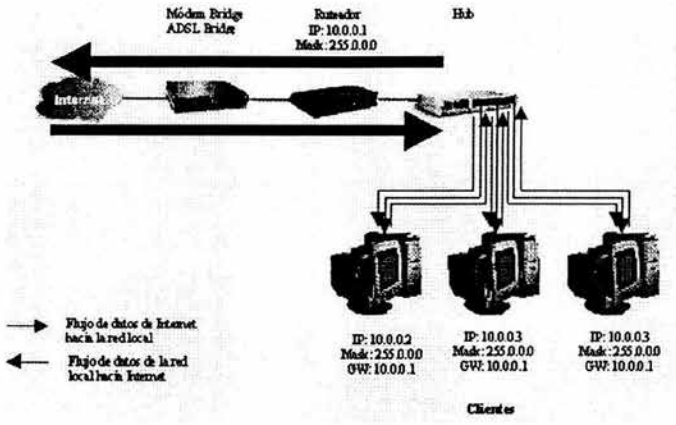


Figura 1.27 Ruteador con NAT configurado

En la figura 1.28 se configura un servidor con NAT para que éste pueda darles a los hosts la salida/entrada a Internet sin necesidad de tener un ruteador. Esto se aplica cuando no se tienen principalmente los recursos económicos y así aprovechar los recursos que se tienen.

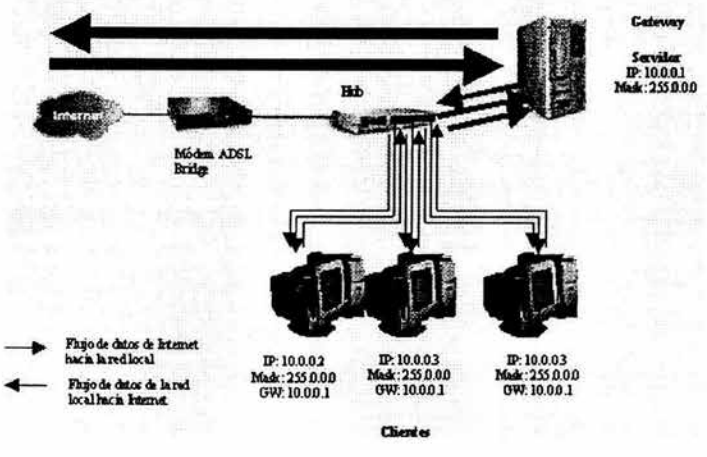


Figura 1.28 Servidor con NAT configurado

1.10 Limitaciones de IPv4

La versión actual de IP (conocida como versión 4 o IPv4) no ha cambiado sustancialmente desde la publicación de RFC 791 en 1981. IPv4 ha demostrado su robustez, facilidad de implementación. Sin embargo, en su diseño original no se contemplaron los siguientes casos:

- El crecimiento exponencial de Internet y el agotamiento del espacio de direcciones IPv4. Esto ha obligado a algunas organizaciones a utilizar el Traductor de direcciones de red (NAT), para asignar múltiples direcciones privadas a una sola dirección IP pública. Aunque NAT permite reutilizar el espacio de direcciones privadas, no admite la seguridad.

- La capacidad de los ruteadores troncales de Internet para mantener grandes tablas de enrutamiento ha llevado a utilizar diferentes mecanismos como: las máscaras de subred de longitud variable (Variable Length Subnet Mask, VLSM), el enrutamiento entre dominios sin clase (Classless Interdomain Routing, CIDR) para el máximo aprovechamiento de direcciones IP.

- La mayor parte de las implementaciones actuales de IPv4 deben configurarse manualmente o mediante un protocolo de configuración de direcciones con estado, como el Protocolo de configuración dinámica de host (Dynamic Host Configuration Protocol, DHCP). Ante un número mayor de equipos y dispositivos que utilizan IP, existe la necesidad de emplear una configuración de direcciones más sencilla y automática.

- En cuestión de seguridad, la comunicación requiere servicios de cifrado que protejan los datos que se envían ante posibles intrusiones o modificaciones durante el tránsito por Internet. Aunque ahora existe un estándar para ofrecer seguridad a los paquetes de IPv4 (conocida como seguridad de Protocolo Internet o IPsec), es opcional.

- La necesidad de facilitar la entrega de datos en tiempo real. Aunque existen estándares de calidad de servicio (Quality of Service, QoS) para IPv4, el tráfico en tiempo real se basa en el campo Tipo de Servicio (Type of Service, TOS) de IPv4 y en la identificación de la carga, normalmente mediante un puerto UDP o TCP. Pero el campo Tipo de Servicio de IPv4 presenta una funcionalidad limitada y con el tiempo han surgido otras implementaciones en tiempo real. Además, la identificación de la carga mediante un puerto TCP y UDP no es posible cuando la carga de paquetes IPv4 está cifrada.

Para resolver estas preocupaciones, el Grupo de trabajo de Ingeniería de Internet (Internet Engineering Task Force, IETF) ha desarrollado un conjunto de protocolos y estándares conocidos como IP versión 6 (IPv6), que incorpora los conceptos de muchos métodos propuestos para actualizar el protocolo IPv4. El diseño de IPv6 se ha diseñado intencionalmente para que afecte lo menos posible a los protocolos de nivel superior e inferior al evitar que se agreguen aleatoriamente nuevas características.

Capítulo 2

Protocolo de Internet
versión 6

CAPITULO 2

Protocolo de Internet Versión 6 (IPv6)

2.1 Historia

Hasta hace algunos años nadie imaginaba que Internet se convertiría en lo es hoy en día: una red de tamaño mundial con un número de usuarios superior a los cien millones y que crece de forma exponencial. La primera Internet creada principalmente con fines experimentales, científico, técnicos y militares, no se parece en nada a la actual, donde ya se advierte una mayor tendencia hacia su comercialización, con mayor número de empresas proveedoras.

Como una solución al crecimiento exponencial de Internet; el Grupo de Trabajo de Ingeniería de Internet (Internet Engineering Task Force, IETF), creó el proyecto **IPng** (de sus siglas en inglés, Internet Protocol the Next Generation), ahora conocido como **IPv6**. Esta nueva versión del Protocolo de Internet sustituirá progresivamente a la actual versión usada en Internet denominada IPv4, ya que IPv6 brinda mejores características que buscan resolver los problemas surgidos con el incremento del uso de las redes de computadoras y el surgimiento de nuevas tecnologías.

IPv6 es la nueva versión del IP que está diseñada como un paso evolutivo de IPv4, y es el resultado de muchas propuestas del IETF y de grupos de trabajo centrados en desarrollar un IPng. El IETF ha producido un conjunto de especificaciones (RFC 1752, 1883, 1886, 1971, 1993, entre otros) que definen la siguiente generación del protocolo IP conocido como " **IPng** " o " **IPv6** "[15].

Estos son los pasos que se han dado en el desarrollo evolutivo de IPv6:

- Para el invierno de 1992 la comunidad del Internet había desarrollado cuatro propuestas diferentes para el IPng que eran: CNAT, IP Encaps, Nimrod y Simple CLNP.

- Después para diciembre del mismo año, aparecieron tres propuestas más el " PIP " (de sus siglas en inglés, The P Internet Protocol), el " SIP " (The Simple Internet Protocol) y el " TP/IX ".

- En la primavera de 1992 el "Simple CLNP" que se desarrolló en el " TUBA" (TCP and UDP with Bigger Addresses ", y el " IP Encaps " en " IPAE " (IP Address Encapsulation).

[15] Hagen Silvia, IPv6 Essentials, p. 1

- Para el verano de 1993, IPAE se combinó con el SIP aunque mantuvo el nombre SIP, que posteriormente se fusionó con la PIPA, y al grupo de trabajo resultante se le llamó "SIPP" (Simple Internet Protocol Plus). Casi al mismo tiempo el grupo de trabajo TP/IX cambió su nombre por el de "CATNIP" (Common Architecture for the Internet)

- Posteriormente, en la reunión del IETF del 25 de julio de 1994 en Toronto Canadá, los directores de área del mismo organismo recomendaron el uso del IPng y lo documentaron en el **RFC 1752**.

- El 17 de noviembre del mismo año fue aprobada esta recomendación por el "IESG" (de sus siglas en inglés, Internet Engineering Steering Group) que elaboró un estándar propuesto.

Hubieron tres fases importantes que influyeron en el desarrollo de IPv6, como se muestra en la figura 2.1

Figura 2.1 Historia IPv6

Como fase adicional se suscitaron varios hechos como la realización de *el 1er. Congreso Internacional de IPv6* (Global IPv6 Summit), organizado por el Foro IPv6 en el año 2000, en Telluride Colorado (Estados Unidos) dando apertura a un ciclo de conferencias "magistrales" sobre los avances del proyecto IPv6. Después de esto, dos cosas relevantes, **Cisco** y **Microsoft** anunciaron sus planes de soportar "oficialmente" IPv6, así mismo, **Sun Microsystem** anunciaba que su sistema

operativo Solaris 8 YA incluía IPv6[16]. De hecho ya existían versiones betas en diversas plataformas que ofrecían dicho soporte.
La nueva versión brinda mejores características en las áreas de direccionamiento, ruteo, aplicaciones en tiempo real y seguridad, puesto que:

- ✓ resuelve los problemas de IPv4 sobre el número de direcciones
- ✓ previene la saturación de tablas de ruteo
- ✓ provee mecanismos de seguridad a nivel capa de red
- ✓ soporta aplicaciones de multimedia y en tiempo real
- ✓ crea los mecanismos de transición para un cambio transparente de IPv4 a IPv6

Es importante mencionar que existieron otras versiones como IPv5 - la versión 5 se asignó al protocolo ST-II - que sólo fue un caso experimental que no tuvo mayor relevancia e IPv7 que fue conocido bajo el titulo de TP/IX (the Next Internet)*.

Para comprender mejor las características de IPv6 se estudiará sus encabezados, primero se revisará el encabezado principal y posteriormente los encabezados de extensión.

2.2 Encabezados de IPv6

El encabezado de IPv6 es una versión optimizada del encabezado de IPv4. Elimina campos innecesarios o que se utilizan muy raramente y agrega otros campos que son más apropiados para el tráfico en tiempo real. Aunque el espacio de direccionamiento de IPv6 es bastante más grande que en la versión 4, el encabezado es solamente dos veces la de dicha versión. Para comprender mejor el encabezado de IPv6 revisar el encabezado de IPv4*. Antes de entrar a ver los encabezados es conveniente ver la estructura de un paquete IPv6 , el cual queda ejemplificada en la figura 2.2.

Figura 2.2 Estructura de un paquete IPv6

[16] Palet Jordi, Tutorial de IPv6, p.6-8
* Para más detalles consultar los RFC´s: 1347, 1475, 1526, 1561, 1707, 1710, 1752, 2373 y 2374.
* veáse Figura 1.8

Encabezado de IPv6

En la figura 2.3 se muestra el encabezado principal IPv6.

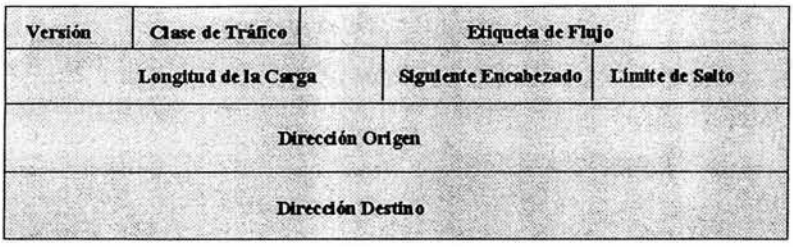


Figura 2.3 Encabezado principal de IPv6

Los campos del encabezado son los siguientes:

Versión (*Version*): se utilizan 4 bits para indicar la versión de IP, que se establece con el valor 6.

Clase de tráfico (*Traffic Class*): indica la clase o la prioridad del paquete IPv6. El tamaño de este campo es de 8 bits. Éste campo proporciona una funcionalidad similar a la del campo Type of Service (Tipo de servicio) de IPv4.

Etiqueta de flujo (*Flow Label*): indica que este paquete pertenece a una secuencia específica de paquetes entre un origen y un destino, lo que requiere un control especial por parte de los *ruteadores* IPv6 intermedios. El tamaño de este campo es de 20 bits. Éste campo se utiliza para conexiones de calidad de servicio, como las que se necesitan para los datos en tiempo real (voz y vídeo).

Longitud de carga (*Payload Length*): indica la longitud de la carga IP. El tamaño de este campo es de 16 bits. Éste campo incluye los encabezados de extensión y la unidad PDU de nivel superior. Con 16 bits, se puede indicar una carga IPv6 de hasta 65.535 bytes. Para longitudes de carga superiores a 65.535 bytes se utiliza la opción de carga Jumbo en el encabezado de extensión Opciones de salto a salto (*Hop-by-Hop Options*).

Siguiente encabezado (*Next Header*): indica el primer encabezado de extensión (si existe) o el protocolo de la unidad PDU de nivel superior (como TCP, UDP o ICMPv6). El tamaño de este campo es de 8 bits. Cuando se indica un protocolo de nivel superior por encima de la capa de Internet, se utilizan aquí los mismos valores que en el campo Protocolo de IPv4.

Límite de salto (*Hop Limit*)**:** indica el número máximo de vínculos por los que puede viajar el paquete IPv6 antes de que se descarte. El tamaño de este campo es de 8 bits. Éste campo es similar al campo TTL de IPv4, excepto en que no existe ninguna relación histórica en cuanto al tiempo (en segundos) que el paquete está en cola en el *ruteador*.

Dirección de origen (*Source Address*): almacena la dirección IPv6 del host de origen. El tamaño de este campo es de 128 bits.

Dirección de destino (*Destination Address):* almacena la dirección IPv6 del host de destino actual. El tamaño de este campo es de 128 bits. Si hay un encabezado de extensión de enrutamiento, la dirección de destino se puede establecer en la interfaz del siguiente *ruteador* de la lista de rutas de origen.

2.2.1 Encabezados de extensión

Puede que no existan o que haya varios encabezados de extensión con distintas longitudes. El campo indica que sigue otro encabezado. El último encabezado de extensión indica el protocolo de nivel superior (como TCP, UDP ó ICMPv6) contenido en la unidad de datos del protocolo de nivel superior.

El encabezado de IPv6 y los encabezados de extensión reemplazan al encabezado de IPv4 con opciones. El formato del nuevo encabezado de extensión permite ampliar IPv6 para que pueda responder a futuras necesidades que ofrezca más capacidades y eficiencia. A diferencia de las opciones del encabezado de IPv4, los encabezados de extensión de IPv6 no tienen un tamaño máximo y pueden ampliarse para aceptar todos los datos de extensión necesarios para la comunicación con IPv6.

En la tabla 2.1 se muestran valores típicos del campo *Siguiente Encabezado* para un encabezado de IPv6 o un encabezado de extensión IPv6.

Valor (en notación decimal)	Encabezado
0	Encabezado Opciones de salto a salto
6	TCP
17	UDP
41	Encabezado de IPv6 encapsulado
43	Encabezado de Enrutamiento
44	Encabezado de Fragmentación
46	Protocolo de reserva de recursos (RSVP)
50	Carga de seguridad de encapsulación
51	Encabezado Autenticación
58	ICMPv6
59	No hay siguiente encabezado
60	Encabezado Opciones de destino

Tabla 2.1 Valores del campo "Next Header"

El encabezado de IPv4 incluye todas las opciones. Por lo tanto, cada ruteador intermedio debe comprobar su existencia y procesarlas cuando están presentes. Esto puede causar un deterioro del rendimiento en el reenvío de paquetes IPv4. Con IPv6, las opciones de entrega y reenvío pasan a los encabezados de extensión. El único encabezado de extensión que debe procesarse en cada ruteador intermedio es el encabezado de extensión Opciones de salto a salto. Así aumenta la velocidad de procesamiento del encabezado de IPv6 y mejora el rendimiento del proceso de reenvío.

Cada encabezado de extensión debe adaptarse a los límites de 64 bits (8 bytes). Los encabezados de extensión de tamaño variable contienen un campo Longitud de extensión de encabezado (*Header Extension Length*) y deben utilizar el relleno cuando sea necesario para asegurarse de que el tamaño sea múltiplo de 8 bytes.

En la figura 2.4 se muestra el campo *Siguiente Encabezado* varios encabezados de extensión que componen una cadena de punteros. Cada puntero indica el tipo de encabezado que viene después del encabezado inmediato hasta que el protocolo de nivel superior se identifica definitivamente.

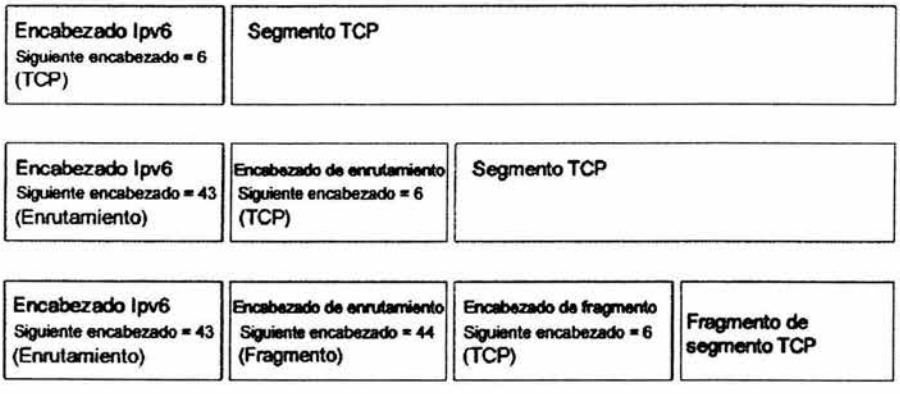


Figura 2.4 Encabezados de extensión de IPv6

Los encabezados de extensión como se muestra la figura 2.5, se procesan en el orden en el que se encuentran. Dado que el único encabezado de extensión procesado por todos los nodos de la ruta de acceso es el encabezado Opciones de salto a salto, debe ser el primero. Hay normas similares para otros encabezados de extensión. En RFC 2460, se recomienda que los encabezados de extensión se coloquen en el encabezado de IPv6 en el orden siguiente:

1. Encabezado Opciones de salto a salto (**Hop-by-Hop Options**).
2. Encabezado Opciones de destino (**Destination Options**).
3. Encabezado de Enrutamiento (**Routing**).

4. Encabezado de Fragmento (**Fragment**).
5. Encabezado de Autenticación (**Authentication**).
6. Encabezado ESP o Carga de seguridad de encapsulación (**Encapsulating Security Payload**).
7. Encabezado Opciones de destino (**Destination Options**).

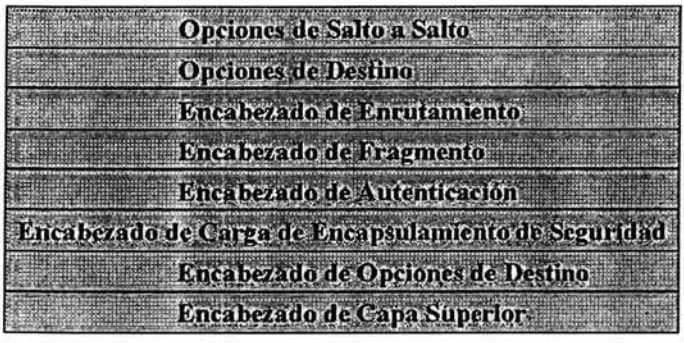


Figura 2.5 Orden de los Encabezados de extensión

Encabezado Opciones de salto a salto (Hop-by-Hop Options)

El encabezado Opciones de salto a salto se utiliza para especificar parámetros de entrega en cada salto de la ruta de acceso al destino.

El encabezado Opciones de salto a salto consta de un campo Siguiente Encabezado, un campo Longitud de extensión del encabezado y un campo Opciones que contiene una o varias opciones. Se utilizan opciones de relleno para garantizar límites de 8 bytes.

Una opción es un encabezado dentro del encabezado de opciones de salto a salto que describe una característica específica de la entrega del paquete o proporciona relleno. Cada opción se codifica en el formato Tipo-Longitud-Valor (TLV), que se utiliza comúnmente en los protocolos TCP/IP. El tipo de opción identifica a la opción y determina el tipo de tratamiento por parte del nodo de procesamiento. La longitud de la opción identifica su longitud. El valor de la opción son los datos asociados a ésta.

Hay algunas opciones que destacan en este encabezado y son las siguientes:

- La opción *Jumbo Payload* (tipo de opción 194) se utiliza para indicar un tamaño de carga superior a 65.535 bytes. Con la opción *Jumbo Payload*, se pueden indicar tamaños de carga de hasta 4.294.967.295 bytes mediante un campo Longitud de Carga Jumbo (*Jumbo Payload Length*) de 32 bits. Un

paquete IPv6 con un tamaño de carga mayor de 65.535 bytes se denomina *jumbograma*.

- La opción Alerta de ruteador (*Router Alert)* se utiliza para indicar al ruteador que el contenido del paquete requiere procesamiento adicional. La opción Alerta de ruteador se utiliza para el Descubrimiento de escucha de multidifusión (Multicast Listener Discovery, MLD) y el Protocolo de reserva de recursos (Resource ReServation Protocol ,RSVP).

Encabezado Opciones de destino (Destination Options)

El encabezado Opciones de destino se utiliza para especificar parámetros de entrega de paquetes para destinos intermedios o para el destino final.

Los campos del encabezado Opciones de destino se definen del mismo modo que el encabezado Opciones de salto a salto.

El encabezado Opciones de destino se utiliza de dos maneras:

1. Si hay un encabezado de enrutamiento, especifica opciones de entrega o de proceso en cada destino intermedio.
2. También especifica opciones de entrega o de proceso en el destino final.

Encabezado de Enrutamiento (Routing)

De forma similar al ruteo de origen que admite IPv4, los nodos de origen de IPv6 pueden utilizar el encabezado de extensión *Enrutamiento* para especificar una ruta de origen, una lista de destinos intermedios para que el paquete viaje por su ruta de acceso al destino final.

El encabezado Enrutamiento consta de un campo Siguiente Encabezado, un campo Longitud de extensión del encabezado (que se define del mismo modo que en el encabezado de extensión Opciones de salto a salto), un campo Tipo de enrutamiento (*Routing Type*), un campo Segmentos restantes (Segments Left) y datos específicos del tipo de enrutamiento.

Los datos específicos del tipo de enrutamiento son una lista de direcciones de destinos intermedios. Cuando el paquete IPv6 llega a un destino intermedio, se procesa el encabezado Enrutamiento y la dirección del siguiente destino intermedio se convierte en la dirección de destino del encabezado de IPv6.

Encabezado de Fragmento (Fragment)

El encabezado Fragmento se utiliza para los servicios de reensamblado y fragmentación de IPv6. El encabezado Fragmento incluye un campo Siguiente Encabezado, un campo Desplazamiento de Fragmentos (Fragment Offset) de 13 bits, un indicador Más fragmentos (More Fragments) y un campo Identificación (Identification) de 32 bits. Los campos Desplazamiento de Fragmentas e Identificación, y el indicador "Más Fragmentos" se utilizan del mismo modo que los

campos correspondientes del encabezado de IPv4. Como el uso del campo Desplazamiento de Fragmentos se define mediante bloques de fragmentos de 8 bytes, el encabezado Fragmento no se puede utilizar para los jumbogramas de IPv6.

En IPv6, sólo los nodos de origen pueden fragmentar las cargas. Si la carga enviada por el protocolo de nivel superior es mayor que la unidad MTU de vínculo o de ruta de acceso, IPv6 fragmenta la carga en el origen y utiliza el encabezado de extensión de Fragmento para proporcionar información de reensamblado.

Cuando se fragmenta un paquete IPv6, se divide inicialmente en una parte que se puede fragmentar y otra parte que no se puede fragmentar.

- La parte que no se puede fragmentar del paquete IPv6 original debe ser procesada por cada nodo intermedio entre el nodo de fragmentación y el destino. Esta parte consta del encabezado de IPv6, el encabezado Opciones de salto a salto, el encabezado Opciones de destino para destinos intermedios y el encabezado de Enrutamiento.
- La parte del paquete IPv6 original que se puede fragmentar sólo debe procesarse en el nodo de destino final. Esta parte consta del encabezado Autenticación, el encabezado Carga de seguridad de encapsulación, el encabezado Opciones de Destino para el destino final y la unidad PDU de nivel superior.

Cada paquete de fragmento consta de la parte que no se puede fragmentar, un encabezado Fragmento y una porción de la parte que se puede fragmentar.

En la figura 2.5 se muestra el proceso de fragmentación para un paquete IPv6.

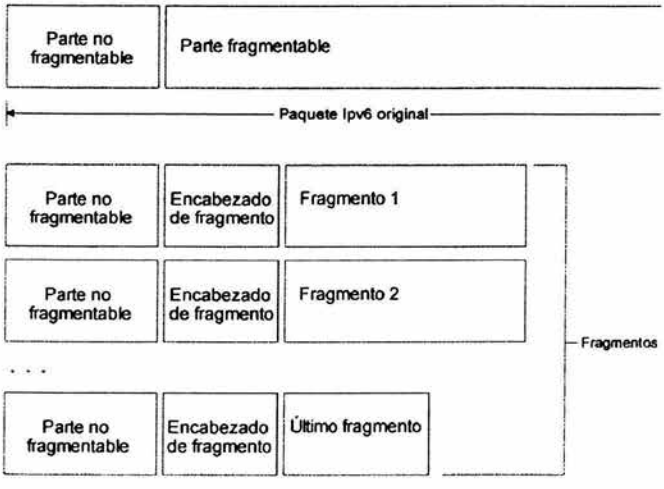


Figura 2.5 Proceso de fragmentación de IPv6

Encabezado Autenticación (Authentication)

El Encabezado Autenticación proporciona autenticación de datos (comprobación del nodo que envió el paquete), integridad de datos (comprobación de que los datos no fueron modificados en el tránsito y protección contra reproducción (garantía de que los paquetes capturados no se pueden volver a transmitir ni ser aceptados nuevamente como datos válidos) para el paquete IPv6.

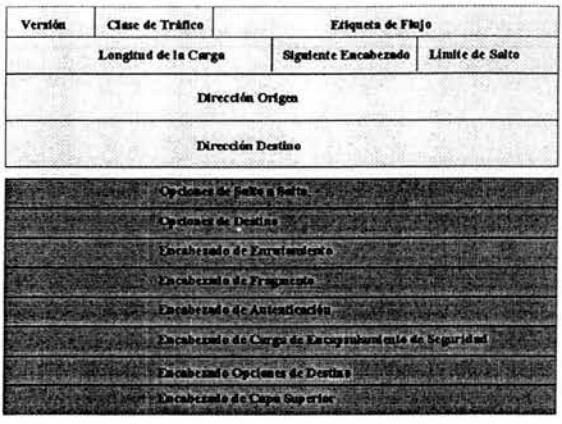
El encabezado Autenticación contiene un campo Siguiente Encabezado, un campo Longitud del encabezado (Header Length), un campo SPI o Índice de parámetros de seguridad (Security Parameters Index) que identifica una asociación de seguridad de seguridad IP (IP Security, IPSec) específica, un campo Número de secuencia (Sequence Number) que proporciona protección contra la reproducción y un campo Datos de autenticación (Autenticación Data) que contiene un valor de comprobación de integridad (Integrity Check Value, ICV). ICV proporciona autenticación de datos e integridad.

El encabezado de extensión Autenticación no proporciona servicios de confidencialidad mediante la encriptación de datos. Para proporcionar esta posibilidad, se puede utilizar el encabezado Autenticación con el encabezado ESP o Carga de seguridad de encapsulación (Encapsulating Security Payload).

Encabezado y finalizador o Carga de seguridad de encapsulación (Encapsulating Security Payload, ESP)

El encabezado y el finalizador Carga de seguridad de encapsulación (Encapsulating Security Payload, ESP) proporcionan servicios de confidencialidad de datos, autenticación de datos e integridad de datos para la carga encapsulada. En cambio, el encabezado Autenticación proporciona servicios de integridad y autenticación de datos para todo el paquete IPv6.

El encabezado ESP contiene un campo o Índice de parámetros de seguridad (Security Parameters Index, SPI) que identifica la asociación de seguridad de IPSec y un campo Número de secuencia (Sequence Number) que proporciona protección contra la reproducción. El finalizador ESP contiene los campos Relleno (Padding), Longitud de relleno (Padding Length), Siguiente Cabecera y Datos de autenticación (Autentication Data). El campo Datos de Autenticación contiene el valor de comprobación de integridad (ICV). En la siguiente figura se muestra todas los encabezados que contiene un paquete de IPv6



. **Figura 2.6** Encabezado de IPv6

En la figura 2.7 se tiene un ejemplo de la captura de un paquete IPv6, donde se muestran los encabezados descritos anteriormente.

Figura 2.7 Captura de un paquete IPv6

Unidad de datos del protocolo de nivel superior

La unidad de datos de protocolo (Protocol Data Unit, PDU) de nivel superior suele constar de un encabezado de protocolo de nivel superior y su carga (por ejemplo, un mensaje ICMPv6, un mensaje UDP o un segmento TCP).

La carga del paquete IPv6 es la combinación de los encabezados de extensión de IPv6 y la unidad PDU de nivel superior. Normalmente, puede tener hasta 65.535 bytes. Las cargas con una longitud superior a los 65.535 bytes se pueden enviar mediante la opción de carga Jumbo en el encabezado de extensión Opciones de salto a salto.

2.3 Direccionamiento

Direccionamiento IPv6

La característica más evidente de IPv6 es el uso de direcciones más grandes. El tamaño de una dirección en IPv6 es de 128 bits, cuatro veces mayor que el de una dirección de IPv4. El espacio de direcciones de 32 bits permite hasta 4.294.967.296 direcciones. Un espacio de direcciones de 128 bits permite hasta 340.282.266.920.938.463.463.374.607.431.768.211.465 (o $3,4 \times 10^{38}$) direcciones.

A finales de la década de 1970, cuando se diseñó el espacio de direcciones de IPv4, era inimaginable que pudiera agotarse. Sin embargo, debido a los cambios tecnológicos y a una práctica de asignaciones en la que no se previó el reciente aumento del número de hosts en Internet, el espacio de direcciones de IPv4 se fue agotando hasta tal punto en que se hizo evidente la necesidad de un reemplazo sustancial.

Para tener una idea algo más aproximada, un espacio de direcciones de 128 bits proporciona 655.570.793.348.866.943.898.599 ($6,5 \times 10^{23}$) direcciones por metro cuadrado de la superficie terrestre de aquí partió la idea de algunos investigadores de asignar direcciones según el área donde se encuentre.

Ciertamente, la decisión de que la dirección de IPv6 tenga una longitud de 128 bits no obedece a que pueda haber hasta $6,5 \times 10^{23}$ direcciones por cada metro cuadrado de la Tierra. El tamaño relativamente grande de una dirección IPv6 se ha diseñado así para que se pueda subdividir en dominios de enrutamiento jerárquico que reflejen la topología de Internet actual. El uso de 128 bits permite varios niveles de jerarquía y ofrece flexibilidad para diseñar un enrutamiento y un direccionamiento jerárquico, algo que actualmente no ofrece la tecnología Internet basada en IPv4*.

Asignación actual

De modo similar al que se utiliza para dividir el espacio de direcciones de IPv4, el espacio de direcciones de IPv6 se divide según el valor de los bits de orden superior. Los bits de orden superior y su valor fijo se conocen como prefijo de formato (Format Prefix, FP).

---

* La arquitectura de direccionamiento IPv6 se describe en RFC 2373

En la tabla 2.1 se muestra la asignación y el reservado del espacio de direcciones de IPv6 por FP

Asignación	Prefijo de formato (FP)	Fracción del espacio de direcciones
Reservado	0000 0000	1/256
Sin asignar	0000 0001	1/256
Reservado para NSAP	0000 001	1/128
Reservado para IPX	0000 010	1/128
Sin asignar	0000 011	1/128
Sin asignar	0000 1	1/32
Sin asignar	0001	1/16
Direcciones de unidifusión global agregables	001	1/8
Sin asignar	010	1/8
Sin asignar	011	1/8
Sin asignar	100	1/8
Sin asignar	101	1/8
Sin asignar	110	1/8
Sin asignar	1110	1/16
Sin asignar	1111 0	1/32
Sin asignar	1111 10	1/64
Sin asignar	1111 110	1/128
Sin asignar	1111 1110 0	1/512
Direcciones de unidifusión de enlace local	1111 1110 10	1/1024
Direcciones de unidifusión de sitio local	1111 1110 11	1/1024
Direcciones de multidifusión	1111 1111	1/256

Tabla 2.1 Asignación del espacio de direcciones de IPv6

El conjunto de direcciones de unidifusión que se pueden utilizar con nodos de IPv6 consta de direcciones de unidifusión global agregables, direcciones de unidifusión de enlace local y direcciones de unidifusión de sitio local. Éstas sólo representan el 15% de todo el espacio de direcciones de IPv6. El 85% restante queda reservado para uso futuro.

2.3.1 Sintaxis de las direcciones de IPv6

Las direcciones de IPv4 se representan en formato de notación decimal con puntos. Esta dirección de 32 bits se divide en límites de 8 bits. Cada conjunto de 8 bits se convierte en su equivalente decimal y está separado por puntos. Para IPv6, **la dirección de 128 bits se divide en límites de 16 bits y cada bloque de 16 bits se convierte en un número hexadecimal de 4 dígitos y se separa con signos de dos puntos (:).** La representación hexadecimal sigue el siguiente esquema:

a) x:x:x:x:x:x:x:x, donde "x" es un valor hexadecimal y cada bloque es de 16 bits. No es necesario escribir los ceros a la izquierda de cada campo.

Por ejemplo, a continuación se muestra una dirección IPv6 en formato binario:

0010000111011010100100001101001100000000010100000010111100111011
0000001010101010000000001111111111111110001010001001110001011010

Esta dirección de 128 bits se divide en límites de 16 bits:

0010000111011010 1001000011010011 **0000000001010000** 0010111100111011
0000001010101010 **0000000011111111** 1111111000101000 **1001110001011010**

Cada bloque de 16 bits se convierte a hexadecimal y está delimitado por signos de dos puntos (:). El resultado es:

21DA:00D3:0000:2F3B:02AA:00FF:FE28:9C5A

b) La representación de IPv6 se puede simplificar aún más si se quitan los ceros a la izquierda de cada bloque de 16 bits. Sin embargo, *cada bloque debe tener un dígito como mínimo.* Al suprimir los ceros a la izquierda, la representación de la dirección se convierte en:

21DA:D3:0:2F3B:2AA:FF:FE28:9C5A

c) Otra forma de representación es la siguiente: **0:0:0:0:0:FFFF:w.x.y.z** ó **::FFFF:w.x.y.z** (donde *w.x.y.z* es la representación decimal con puntos de una dirección IPv4), que es utilizado para representar un nodo que es sólo de IPv4 ante un nodo IPv6. Se utiliza únicamente para la representación interna. La dirección asignada de IPv4 nunca se utiliza como dirección de origen o de destino de un paquete IPv6. Ejemplo:

0:0:0:0:0:FFFF:132.248.10.3

se puede representar como:

::FFFF:132.248.10.3

Y cuando tenemos un entorno mixto IPv4 e IPv6, se representa **0:0:0:0:0:0:w:x:y:z** ó **::w.x.y.z**; este formato es utilizado por nodos de doble pila (Dual Stack) que se comunican con IPv6 sobre una infraestructura de IPv4. Los nodos de doble pila son nodos con protocolos IPv4 e IPv6. Cuando se utiliza la dirección compatible con IPv4 como destino de IPv6, el tráfico de IPv6 se encapsula automáticamente con un encabezado de IPv4 y se envía al destino mediante la infraestructura de IPv4. Ejemplo:

0:0:0:0:0:0:132.248.120.4

se puede representar como:

::132.248.120.4

Compresión de ceros

Algunos tipos de direcciones contienen secuencias de ceros. Para simplificar aún más la representación de direcciones de IPv6, *una secuencia contigua de bloques* de 16 bits establecida como 0 en formato hexadecimal con dos puntos se puede comprimir como "**::**".

Por ejemplo, la dirección de enlace local de FE80:0:0:0:2AA:FF:FE9A:4CA2 se puede comprimir en FE80::2AA:FF:FE9A:4CA2. Ahora bien la dirección de multidifusión FF02:0:0:0:0:0:0:2 se puede comprimir en FF02::2.

La compresión de cero sólo se puede utilizar para comprimir una serie contigua de bloques de 16 bits expresada en notación hexadecimal con dos puntos. No se puede utilizar la compresión de ceros para incluir una parte de un bloque de 16 bits. Por ejemplo, no se puede expresar FF02:**30**:0:0:0:0:0:5 como FF02:3::5.

Para determinar cuántos bits 0 se representan mediante "**::**",se debe contar el número de bloques de la dirección comprimida, restar ese número a 8 y multiplicar el resultado por 16. Por ejemplo, en la dirección FF02::2, hay dos bloques (el bloque "FF02" y el bloque "2"). El número de bits expresado por "**::**" es 96 (96 = (8 – 2)*16).

La compresión de ceros sólo se puede utilizar una vez en una dirección dada. De lo contrario, no se podría determinar el número de bits 0 representados por cada instancia de "**::**".

Otros ejemplos:

1090:0:0:0:9:800:300B:420C (dirección unicast)
FF01:0:0:0:0:0:0:102 (dirección multicast)
0:0:0:0:0:0:0:1 (dirección loopback)

pueden representarse como:

1090::9:800:300B:420C (dirección unicast)
FF01::102 (dirección multicast)
::1 (dirección loopback)

Prefijos IPv6

El prefijo es la parte de la dirección que indica **los bits con valores fijos** o los bits del identificador de red. Un prefijo IPv6 se escribe con la notación:

dirección-IPv6/longitud-del-prefijo

Por ejemplo, FE80::2AA:FF:FE9A:4CA2/64 indica que los primeros 64 bits de la dirección corresponden al prefijo de red. La notación de prefijo también *se utiliza para expresar los identificadores de red o de subred.* Por ejemplo, 21DA:D3::/48 es una subred.

Una dirección de nodo, con su prefijo, se puede utilizar para obtener el identificador de subred. Por ejemplo, el identificador de subred derivado de la dirección y el prefijo 21DA:D3:0:2F3B:2AA:FF:FE28:9C5A/64 es :

21DA:D3:0:2F3B::/64.

Las direcciones de IPv4 suelen utilizar una representación decimal con puntos del prefijo de red, que se conoce como máscara de subred*.Para IPv6 **no se utiliza la máscara de subred**. Sólo se admite la notación de longitud de prefijo.

Aunque se pueden definir prefijos a lo largo de los límites de bit, la notación hexadecimal con dos puntos para las direcciones IPv6 se expresa a lo largo de límites de cuarteto (4 bits). Para expresar correctamente una subred con un prefijo cuya longitud no es múltiplo de 4, deberá realizar conversiones de notación hexadecimal a binaria para determinar el identificador de subred adecuado. Por ejemplo, para expresar la subred de la dirección y el prefijo de 21DA:D3:0:2F3B:2AA:FF:FE28:9C5A/59, deberá convertir el "3" de "2F3B" a binario (0011), dividir el cuarteto entre el tercer y el cuarto dígito binario, y volver a realizar la conversión a hexadecimal. El resultado es el identificador de subred 21DA:D3:0:2F20::/59.

2.3.2 Tipos de direcciones IPv6

Hay tres tipos de direcciones IPv6:

1. *Unidifusión* ***(Unicast)***

* Veáse Máscara de subred, capítulo 1.4.2.4

Una dirección de unidifusión identifica a una sola interfaz en el ámbito del tipo de dirección de unidifusión. Los paquetes dirigidos a una dirección de unidifusión se entregan a una sola interfaz. También permite que varias interfaces utilicen la misma dirección, siempre y cuando las distintas interfaces aparezcan como una sola interfaz para la implementación de IPv6 en el host.

2. *Cualquier difusión* ***(Anycast)***
   Una dirección para cualquier difusión identifica a varias interfaces. Los paquetes dirigidos a una dirección para cualquier difusión se entregan a una sola interfaz, la más próxima que identifica la dirección. La interfaz "más próxima" se define como la más cercana en términos de distancia de enrutamiento. Una dirección para cualquier difusión se utiliza para la comunicación "**de uno a uno de muchos**", con entrega a una sola interfaz.

3. *Multidifusión* ***(Multicast)***
   Una dirección de multidifusión identifica a varias interfaces. Los paquetes dirigidos a una dirección de multidifusión se entregan a todas las interfaces identificadas por la dirección. Una dirección de multidifusión se utiliza para la comunicación "**de uno a muchos**", con entrega a varias interfaces.

Es importante recalcar que en todos los casos, las direcciones IPv6 identifican interfaces, no nodos. Un nodo se identifica mediante cualquier dirección de unidifusión asignada a una de sus interfaces.

2.3.2.1 Direcciones IPv6 de unidifusión (unicast)

Los siguientes tipos de direcciones son direcciones IPv6 de unidifusión:

- Direcciones de unidifusión global agregables
- Direcciones de enlace local
- Direcciones de sitio local
- Direcciones especiales
- Direcciones NSAP e IPX

Direcciones de unidifusión global agregables

Las direcciones de unidifusión global agregables, identificadas mediante el *Format Prefix* 001 (visto en la tabla 2.1), equivalen a las direcciones IPv4 públicas. Se pueden *rutear* globalmente y es posible el acceso a las mismas en la parte de IPv6 de Internet, conocida como 6bone (red troncal de IPv6).

Las direcciones de unidifusión global agregables están diseñadas para ser agregadas de modo que se obtenga una infraestructura de enrutamiento eficiente.

En la figura 2.7 se muestra la estructura de una dirección de unidifusión global agregable

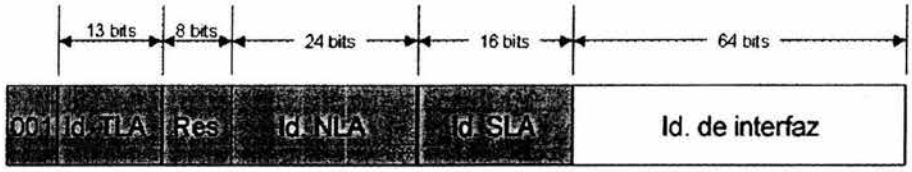


Figura 2.7 Dirección de unidifusión global agregable

Los campos de la dirección de unidifusión global agregable son:

TLA (Id. de TLA): indica el Agregador de Nivel Superior (Top Level Aggregation, TLA). El tamaño de este campo es de 13 bits. El TLA identifica el nivel superior de la jerarquía de enrutamiento. La asociación IANA administra los TLA, que se asignan a registros locales de Internet que, a su vez, asignan TLA individuales a grandes proveedores de servicios Internet (Internet Service Providers, ISP) de largo alcance. Los ruteadores del nivel superior de la jerarquía de enrutamiento en Internet de IPv6, sólo tienen rutas con prefijos de 16 bits que corresponden a los TLA asignados.

Res: bits reservados para uso futuro al expandir el tamaño del Id. de TLA o del Id. de NLA. El tamaño de este campo es de 8 bits.

NLA (Id. de NLA): indica el Agregador del Siguiente Nivel (Next-Level Aggregation, NLA). El Id. de NLA se utiliza para identificar un sitio de cliente específico. El tamaño de este campo es de 24 bits. El Id. de NLA permite a un ISP crear varios niveles de jerarquía de direccionamiento dentro de una red para organizar el enrutamiento y el direccionamiento de los ISP en un nivel inferior e identificar sitios.

SLA (Id. de SLA): indica el Agregador de Nivel de Sitio (Site-Level Aggregation, SLA). El Id. de SLA puede servir a una organización para identificar subredes dentro de su sitio. El tamaño de este campo es de 16 bits. La organización puede utilizar estos 16 bits en su sitio para crear 65.536 subredes (una clase A de IPv4) o niveles múltiples de jerarquía de direccionamiento.

Interface (Id. de interfaz): indica la interfaz de una subred específica. El tamaño de este campo es de 64 bits.

Los campos de una dirección de unidifusión global agregable crean la estructura en tres niveles que se muestra en la figura 2.8.

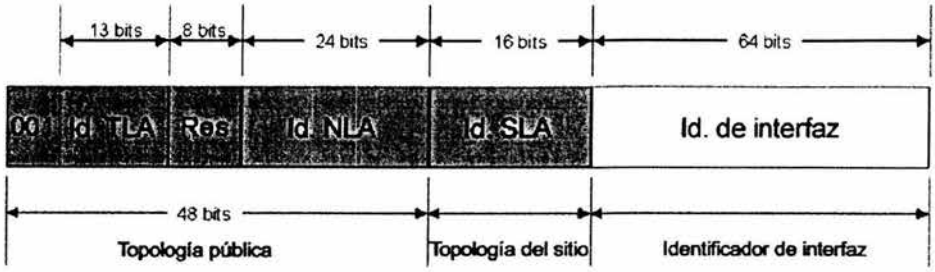


Figura 2.8 Estructura en tres niveles de la dirección de unidifusión global agregable

La topología pública es la colección de ISP's grandes y pequeños que proporcionan acceso a la parte IPv6 de Internet. La topología del sitio es la colección de subredes del sitio de una organización. El identificador de interfaz identifica a una interfaz específica de una subred en el sitio de una organización*.

Direcciones de unidifusión de uso local

Hay dos tipos de direcciones de unidifusión de uso local:

1. Direcciones de enlace local utilizadas entre vecinos de enlace y para procesos *Neighbor Discovery*.
2. Direcciones de sitio local utilizadas entre nodos que se comunican con otros nodos del mismo sitio.

Direcciones de enlace local

Los nodos utilizan las direcciones de enlace local identificadas mediante *FP 1111 1110 10* (ver tabla 2.1) cuando se comunican con nodos vecinos en el mismo enlace. Por ejemplo, en una red IPv6 de enlace único sin ruteador, las direcciones de enlace local se utilizan para la comunicación entre los hosts. El ámbito de una dirección de enlace local es el enlace local.

Se necesita una dirección de enlace local para los procesos de descubrimiento de vecino (Neighbor Discovery, ND) y siempre se configura automáticamente, incluso en ausencia de todas las demás direcciones de unidifusión.

En la figura 2.9 se muestra la estructura de la dirección de enlace local.

* Para obtener más información acerca de las direcciones de unidifusión global agregables, consulte RFC 2374.

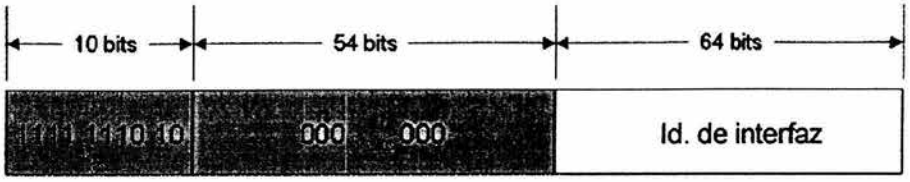


Figura 2.9 Dirección de enlace local

Las direcciones de enlace local siempre empiezan por **FE80**. Con el identificador de interfaz de 64 bits, el prefijo para las direcciones de enlace local es siempre FE80::/64. Un ruteador IPv6 nunca reenvía el tráfico de enlace local más allá del enlace.

Direcciones de sitio local

Las direcciones de sitio local, identificadas mediante FP 1111 1110 11(ver tabla 2.1), equivalen al espacio de direcciones no homologadas de IPv4 **(10.0.0.0/8, 172.16.0.0/12 y 192.168.0.0/16)**. Por ejemplo, las intranets privadas que no tienen una conexión directa a Internet de IPv6 pueden utilizar direcciones de sitio local sin entrar en conflicto con direcciones de unidifusión global agregables. No se puede tener acceso a las direcciones de sitio local desde otros sitios y los ruteadores no deben reenviar el tráfico local fuera del sitio. Las direcciones de sitio local se pueden utilizar junto con las direcciones de unidifusión global agregables. El ámbito de una dirección de sitio local es el sitio (la red interna de la organización).

A diferencia de las direcciones de enlace local, las direcciones de sitio local no se configuran automáticamente y deben asignarse a través de procesos de configuración de direcciones sin estado (stateless) y con estado (Stateful).

En la figura 2.10 se muestra la estructura de la dirección de sitio local.

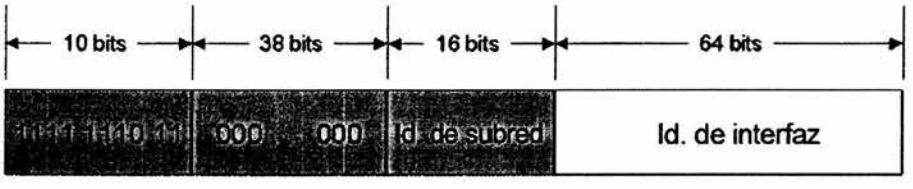


Figura 2.10 Dirección de sitio local

Los primeros 48 bits son siempre fijos para las direcciones de sitio local, que empiezan por FEC0::/48. Después de los 48 bits fijos hay un identificador de subred de 16 bits (Id. de subred) que proporciona 16 bits, con el que se pueden crear subredes en una organización y se pueden subdividir los bits de orden superior del campo Id. de subred para crear una infraestructura de enrutamiento

agregable y jerárquica. Después del campo Id. de subred hay un campo Id. de interfaz que identifica una interfaz específica en una subred.

Direcciones IPv6 especiales

A continuación se muestran direcciones IPv6 especiales:

- Dirección no especificada
  La dirección no especificada (0:0:0:0:0:0:0:0 ó ::) sólo se utiliza para indicar la ausencia de una dirección. Equivale a la dirección IPv4 no especificada 0.0.0.0. La dirección no especificada no se asigna nunca a una interfaz ni se utiliza como dirección de destino.
- Dirección de bucle de retroceso (loopback)
  La dirección loopback (0:0:0:0:0:0:0:1 ó ::1) se utiliza para identificar una interfaz de bucle de retroceso, lo que permite que un nodo se envíe paquetes a sí mismo. Equivale a la dirección loopback de IPv4: 127.0.0.1. Los paquetes dirigidos a la dirección de bucle de retroceso nunca deben enviarse a través de un enlace o reenviarse mediante un ruteador de IPv6.

Direcciones NSAP e IPX

Para proporcionar un medio de asignar direcciones de Punto de Acceso a Servicios de Red (Network Service Access Point, NSAP) y de Intercambio de paquetes entre redes (Internetwork Packet Exchange, IPX) a direcciones IPv6, se definen direcciones NSAP e IPX.

- Dirección IP
  Las direcciones NSAP utilizan FP 0000001 y asignan los últimos 121 bits de la dirección IPv6 a una dirección NSAP*.
- Direcciones IPX
  Las direcciones IPX utilizan FP 0000010 y asignan los últimos 121 bits de la dirección IPv6 a una dirección IPX. Aún no se ha definido la asignación de una dirección IPX a una dirección IPv6.

2.3.2.2 Direcciones IPv6 para cualquier difusión (anycast)

Una dirección para cualquier difusión se asigna a varias interfaces. La infraestructura de enrutamiento reenvía los paquetes dirigidos a una dirección de unidifusión a la interfaz más próxima a la que esté asignada la dirección para cualquier difusión. Para facilitar la entrega, la infraestructura de enrutamiento debe conocer las interfaces a las que se asignan direcciones para cualquier difusión y su "distancia" en términos de medida de enrutamiento.

---

* Para obtener más información acerca de los cuatro tipos de asignaciones de direcciones NSAP, consulte RFC 1888

Actualmente, las direcciones para cualquier difusión sólo se utilizan como direcciones de destino y se asignan únicamente a los *ruteadores*. Las direcciones para cualquier difusión se asignan fuera del espacio de direcciones de unidifusión y el ámbito de una dirección para cualquier difusión es el ámbito del tipo de dirección de unidifusión desde el que se asigna la dirección para cualquier difusión.

La dirección para cualquier difusión se crea a partir del prefijo de subred para una interfaz dada. Para crear la dirección para cualquier difusión, los bits del prefijo de subred quedan fijos en sus valores correspondientes y los bits restantes se establecen en 0. La figura 2.11 ilustra la dirección para cualquier difusión.

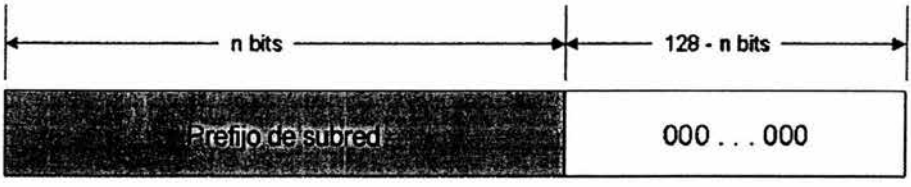


Figura 2.11 Dirección para cualquier difusión.

Todas las interfaces de ruteador conectadas a una subred se asignan a la dirección para cualquier difusión de la subred. La dirección para cualquier difusión se utiliza para la comunicación con uno o varios *ruteadores* conectados a una subred remota.

Por lo general, un host IPv4 con un sólo adaptador de red tiene una única dirección IPv4 asignada al adaptador. Sin embargo, un host IPv6 suele tener varias direcciones IPv6, incluso con una sola interfaz. A un host IPv6 se le asignan las siguientes direcciones de unidifusión:

- Una dirección de enlace local para cada interfaz.
- Direcciones de unidifusión para cada interfaz (que podrían ser una dirección de sitio local y una o varias direcciones de unidifusión global agregables).
- Una dirección de loopback (::1).

Un host IPv6 típico puede tener varias interfaces o direcciones porque tiene al menos dos direcciones con las que puede recibir paquetes (una dirección de enlace local para el tráfico local y una dirección agregable o de sitio local que se puede rutear).

Además, cada host escucha el tráfico en las siguientes direcciones de multidifusión:

- La dirección de multidifusión de todos los nodos del ámbito local de nodo (FF01::1).

- La dirección de multidifusión de todos los nodos del ámbito de enlace local (FF02::1).
- La dirección de nodo solicitado para cada dirección de unidifusión.
- Las direcciones de multidifusión de los grupos unidos.

Ahora bien, para un *ruteador* IPv6 se le asignan las siguientes direcciones:

- Una dirección de enlace local para cada interfaz.
- Direcciones de unidifusión para cada interfaz (que podrían ser una dirección de sitio local y una o varias direcciones de unidifusión global agregables).
- Una dirección para cualquier difusión.
- Direcciones adicionales para cualquier difusión (opcional).
- Una dirección de loopback (::1).

Además, cada ruteador escucha el tráfico en las siguientes direcciones de multidifusión:

- La dirección de multidifusión de todos los nodos del ámbito local de nodo (FF01::1).
- La dirección de multidifusión de todos los ruteadores del ámbito local de nodo (FF01::2).
- La dirección de multidifusión de todos los nodos del ámbito de enlace local (FF02::1).
- La dirección de multidifusión de todos los ruteadores del ámbito de enlace local (FF02::1).
- La dirección de multidifusión de todos los ruteadores del ámbito de sitio local (FF05::2).
- La dirección de nodo solicitado para cada dirección de unidifusión.

2.3.2.3 Direcciones IPv6 de multidifusión (multicast)

En IPv6, el tráfico de multidifusión funciona del mismo modo que en IPv4. Los nodos IPv6 ubicados arbitrariamente pueden atender al tráfico de multidifusión en una dirección de multidifusión IPv6 arbitraria. Los nodos IPv6 pueden escuchar a varias direcciones de multidifusión simultáneamente. Los nodos pueden unirse a un grupo de multidifusión o abandonarlo en cualquier momento.

Las direcciones de multidifusión utilizan FP 11111111 (ver tabla 2.1). Es fácil clasificar una dirección IPv6 como de multidifusión, ya que siempre empieza por "**FF**". Las direcciones de multidifusión no se pueden utilizar como direcciones de origen o como destinos intermedios en un encabezado de enrutamiento.

Además de FP, las direcciones de multidifusión incluyen una estructura adicional para identificar sus indicadores, ámbito y grupo de multidifusión. En la figura 2.12 se muestra la dirección de multidifusión IPv6.

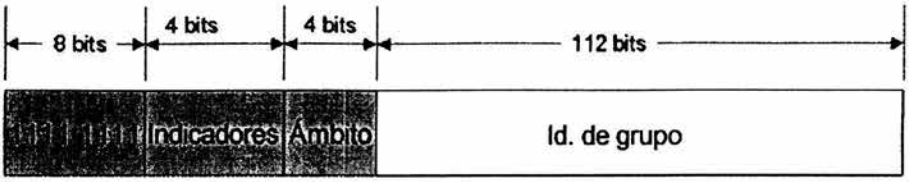


Figura 2.12 Dirección de multidifusión IPv6.

Dirección de nodo solicitado

La dirección de nodo solicitado facilita una consulta eficiente de los nodos de red durante la resolución de direcciones. En IPv4, la trama de solicitud de ARP se envía a la difusión (broadcast) a nivel de MAC, lo que afecta a todos los nodos del segmento de red, incluidos los que no utilizan IPv4. IPv6 utiliza el mensaje Descubrimiento de vecino (Neighbor Discovery, ND) para realizar la misma operación. Sin embargo, en vez de utilizar la dirección de multidifusión de todos los nodos de ámbito de enlace local como destino del mensaje Solicitud de vecino (Neighbor Solicitation, NS), lo que afectaría a todos los nodos IPv6 del enlace local, se utiliza la dirección de multidifusión de nodo solicitado. La dirección de multidifusión de nodo solicitado consta del prefijo FF02::1:FF00:0/104 y los últimos 24 bits de la dirección IPv6 que se va a resolver.

Por ejemplo, para el nodo con la dirección IPv6 de enlace local, se tiene:

FE80::2AA:FF:FE28:9C5A, la dirección de nodo solicitado correspondiente es FF02::1:FF28:9C5A. Para resolver la dirección FE80::2AA:FF:FE28:9C5A en la dirección de su nivel de enlace, un nodo puede enviar un mensaje *Neighbor Solicitation* a la dirección de nodo solicitado FF02::1:FF28:9C5A. El nodo que utiliza la dirección FE80::2AA:FF:FE28:9C5A escucha el tráfico de multidifusión en la dirección del nodo solicitado y, para las interfaces que corresponden a una tarjeta adaptadora de red física, habrá registrado la dirección de multidifusión correspondiente con la tarjeta adaptadora de red.

El resultado del uso de la dirección de multidifusión de nodo solicitado es que, para las resoluciones de direcciones, algo que ocurre comúnmente en los enlaces, no se necesita un mecanismo que afecte a todos los nodos de la red. Si se utiliza la dirección de nodo solicitado, muy pocos nodos se ven afectados durante la resolución de direcciones. En la práctica, debido a la relación existente entre la dirección MAC de Ethernet, el Id. de interfaz y la dirección de nodo solicitado, la dirección de nodo solicitado actúa como dirección de pseudo-unidifusión para una resolución de direcciones eficiente.

Asignando direcciones de multidifusión IPv6 a direcciones Ethernet

Cuando envía paquetes de multidifusión IPv6 a través de un enlace Ethernet, la dirección MAC de destino es 33-33-mm-mm-mm-mm, donde mm-mm-mm-mm es una asignación directa de los últimos 32 bits de la dirección de multidifusión IPv6.

Para recibir de un modo eficiente paquetes de multidifusión IPv6 a través de un enlace Ethernet, los adaptadores de red Ethernet pueden almacenar otras direcciones MAC de interés en una tabla del adaptador de red. Si se recibe una trama Ethernet con una dirección MAC de interés, se pasa a las capas superiores para su procesamiento. Para todas las direcciones de multidifusión que escucha el host, existe una entrada correspondiente en la tabla de direcciones MAC de interés.

Por ejemplo, un host con la dirección MAC Ethernet 00-AA-00-3F-2A-1C (dirección de enlace local FE80::2AA:FF:FE3F:2A1C) registra las siguientes direcciones MAC de multidifusión con el adaptador Ethernet:

- La dirección 33-33-00-00-00-01, que corresponde a la dirección de multidifusión de todos los nodos de ámbito de enlace local FF02::1.
- La dirección 33-33-FF-3F-2A-1C, que corresponde a la dirección de nodo solicitado FF02::1:FF3F:2A1C. Recuerde que la dirección de nodo solicitado se compone del prefijo FF02::1:FF00:0/104 y los últimos 24 bits de la dirección IPv6 de unidifusión.

Según sea necesario, se agregan o se quitan direcciones de multidifusión adicionales de la tabla de direcciones de interés del adaptador de red Ethernet en el host que escucha.

2.3.3 Direcciones EUI-64

Todas las direcciones que utilizan los prefijos comprendidos entre 001 y 111 deben utilizar también un identificador de interfaz de 64 bits que está derivado de la dirección EUI-64. La dirección EUI-64 de 64 bits fue definida por el Instituto de Ingeniería Eléctrica y Electrónica (Institute of Electrical and Electronic Engineers, IEEE). Las direcciones EUI-64 se asignan a una tarjeta adaptadora de red o se derivan de direcciones IEEE 802.

Direcciones IEEE 802

Los identificadores de interfaz tradicionales de los adaptadores de red utilizan una dirección de 48 bits denominada dirección IEEE 802. Consta de un *Id. de compañía* de 24 bits (también conocido como Identificador del fabricante) y un *Id. de extensión* de 24 bits (también conocido como Identificador de tarjeta). La combinación del *Id. de compañía*, que se asigna en exclusiva a cada fabricante de adaptadores de red, y el *Id. de tarjeta*, que se asigna en exclusiva a cada

adaptador de red en el momento del montaje, genera una dirección exclusiva global de 48 bits. Esta dirección de 48 bits también se denomina dirección física, de hardware o de control de acceso a medios (Media Access Control, MAC).

En la figura 2.13 se muestra la estructura de la dirección IEEE 802 de 48 bits.

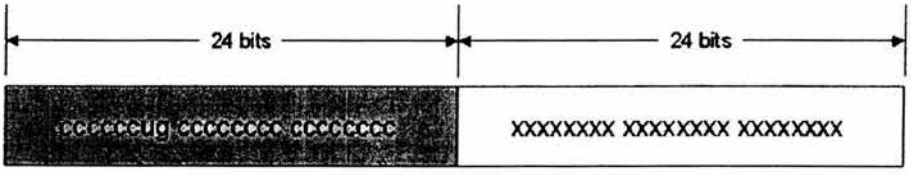


Figura 2.13 Dirección IEEE 802 de 48 bits.

Los bits definidos en la dirección IEEE 802 son:

Universal/Local (U/L): el bit situado junto al bit de orden inferior en el primer byte se utiliza para indicar si la dirección se administra universal o localmente. Si el bit U/L está establecido en el valor 0, IEEE ha administrado la dirección a través de la designación de un Id. de compañía. Si el bit U/L está establecido en el valor 1, la dirección se administra localmente. El administrador de la red ha suplantado la dirección del fabricante y ha especificado otra dirección. El bit U/L bit se designa mediante **u** en la figura 2.13.

Individual/Group (I/G) (Individual/Grupo): el bit de orden inferior del primer byte se utiliza para indicar si se trata de un dirección individual (de unidifusión) o de grupo (de multidifusión). Cuando está establecido en el valor 0, la dirección es de unidifusión. Cuando está establecido en el valor 1, la dirección es de multidifusión. El bit I/G se designa mediante **g** en la figura 2.13.

Para una dirección de adaptador de red 802.x típica, tanto el bit U/L como el bit I/G se establecen en el valor 0, que corresponde a una dirección MAC de unidifusión administrada de forma universal.

Identificadores de interfaz IEEE EUI-64

La dirección IEEE EUI-64 representa un estándar en el direccionamiento de interfaces de red. El *Id. de la compañía* también tiene 24 bits, pero el *Id. de extensión* es de 40 bits, lo que representa un espacio de direcciones mucho mayor para el fabricante de adaptadores de red. La dirección EUI-64 utiliza los bits U/L e I/G del mismo modo que la dirección IEEE 802.

En la figura 2.14 se muestra la estructura de la dirección EUI-64.

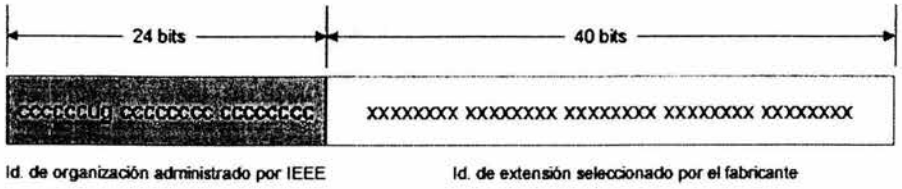


Figura 2.14 Dirección EUI-64

Asignar direcciones IEEE 802 a direcciones EUI-64

Para crear una dirección EUI-64 a partir de una dirección IEEE 802, los 16 bits de 11111111 11111110 (0xFFFE) se insertan en la dirección IEEE 802 entre el Id. de la compañía y el Id. de extensión, tal como se muestra en la figura 2.15

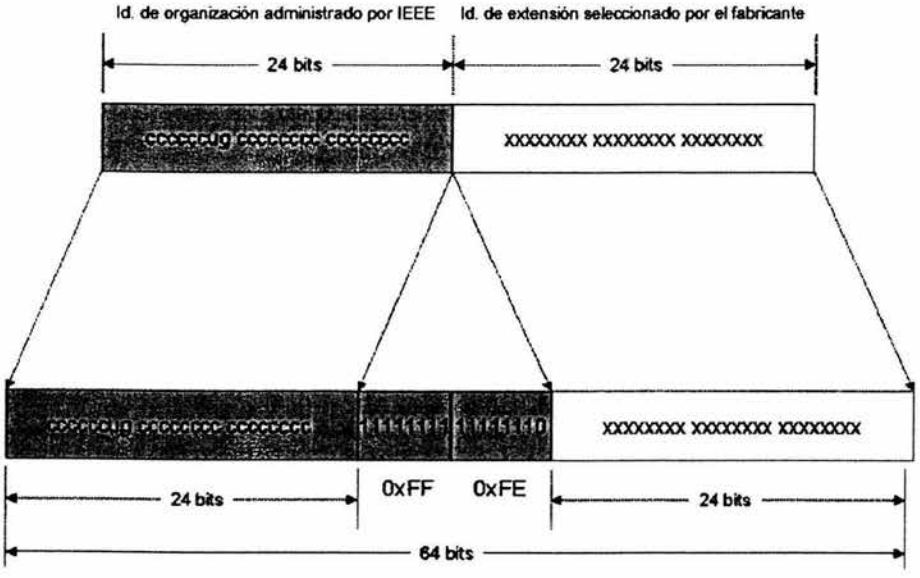


Figura 2.15 Conversión de una dirección IEEE 802 en una dirección EUI-64

Para obtener el identificador de interfaz de 64 bits para direcciones de unidifusión IPv6, el bit U/L de la dirección EUI-64 se complementa. En la figura 2.16 se muestra la conversión de una dirección EUI-64 de unidifusión administrada universalmente.

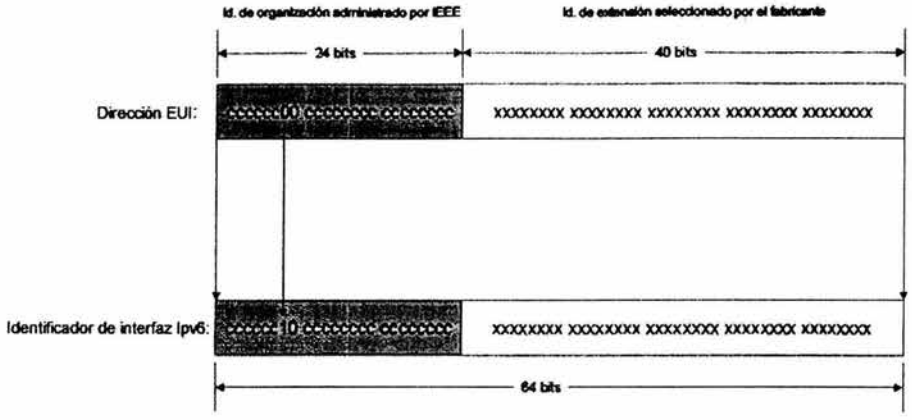


Figura 2.16 Conversión de una dirección EUI-64 de unidifusión administrada universalmente en un identificador de interfaz de IPv6.

Para obtener un identificador de interfaz de IPv6 a partir de una dirección IEEE 802, en primer lugar deberá asignar la dirección IEEE 802 a una dirección EUI-64 y después, complementar el bit U/L. En la figura 2.17 se muestra el proceso de conversión de una dirección IEEE 802 de unidifusión administrada universalmente.

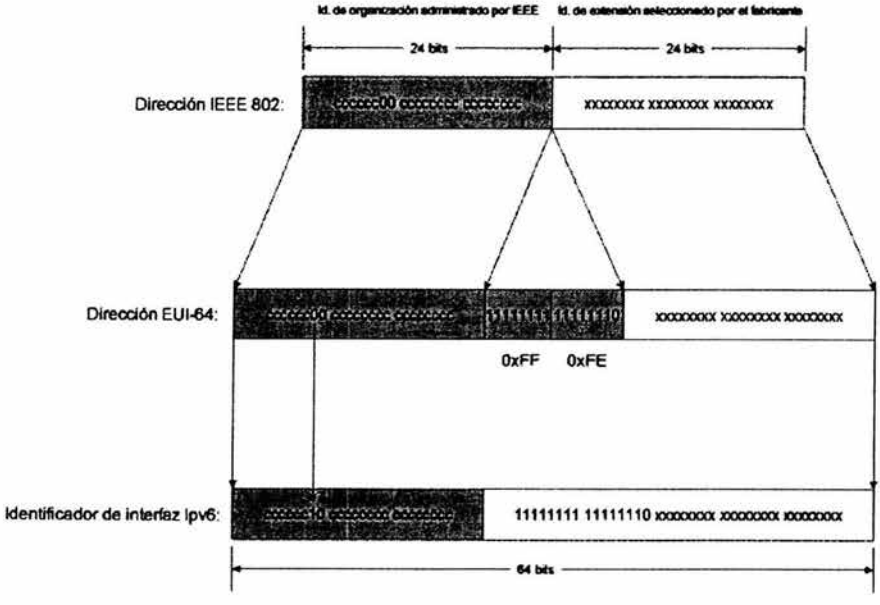


Figura 2.17 Conversión de una dirección IEEE 802 de unidifusión administrada universalmente en identificador de interfaz de IPv6.

A continuación se da un ejemplo de una conversión usando la dirección IEEE 802.

El Host A tiene la dirección MAC Ethernet 00-AA-00-3F-2A-1C. En primer lugar, se convierte al formato EUI-64 por la inserción de **FF-FE** entre el tercer y el cuarto bytes, lo que genera 00-AA-00-FF-FE-3F-2A-1C. A continuación, el bit U/L, que es el séptimo del primer byte, se complementa. El primer byte en forma binaria es 00000000. Cuando se complementa el séptimo bit, se convierte en 00000010 (0x02). El resultado final es 02-AA-00-FF-FE-3F-2A-1C que, cuando se convierte a la notación hexadecimal con puntos, pasa a ser el identificador de interfaz 2AA:FF:FE3F:2A1C. Como resultado*, la dirección de enlace local que corresponde al adaptador de red con la dirección MAC 00-AA-00-2A-1C es FE80::2AA:FF:FE3F:2A1C.

2.4 Protocolo de Mensajes de Control de Internet versión 6 (ICMPv6)

Ahora IPv6 utiliza una versión actualizada del Protocolo de mensajes de control de Internet (*Internet Control Message Protocol,* ICMP), denominado ICMP versión 6 (ICMPv6). ICMPv6 presenta las funciones comunes de ICMP IPv4 acerca de errores de entrega o reenvío y proporciona un servicio de eco simple para la solución de problemas[17].

El protocolo ICMPv6 también proporciona lo siguiente:

- Descubrimiento de escucha de multidifusión (Multicast Listener Discovery, MLD)

  MLD es un conjunto de tres mensajes ICMP que reemplazan a la versión 2 del Protocolo de administración de grupos de Internet (IGMP) de IPv4 ya que administra la pertenencia a grupos de multidifusión de subred.

- Descubrimiento de vecino (Neighbor Discovery, ND)

  Neighbor Discovery es un conjunto de cinco mensajes ICMPv6 que administran la comunicación entre nodos en un vínculo. Neighbor Discovery reemplaza al Protocolo de resolución de direcciones (ARP), al proceso Router Discovery (Descubrimiento de ruteadores) de ICMPv4 y al mensaje Redirect (Redirección) de ICMPv4.

---

* Cuando complemente el bit U/L, agregue 0x2 al primer byte si la dirección se administra universalmente y reste 0x2 del primer byte si la dirección se administra localmente.

[17] Miller Mark, Implementing IPv6, p. 108, 110

Tipos de mensajes ICMPv6

Hay dos tipos de mensajes ICMPv6:

1. Mensajes de error

   Los mensajes de error se utilizan para informar de la existencia de errores en el reenvío o en la entrega de paquetes IPv6 por parte del nodo de destino o de un ruteador intermedio. Los mensajes de error ICMPv6 son ***Destino inaccesible*** (Destination Unreachable), ***Paquete demasiado grande*** (Packet Too Big), ***Fin de tiempo de espera*** (Time Exceeded) y ***Problema de parámetro*** (Parameter Problem).

2. Mensajes informativos

   Los mensajes informativos se utilizan para proporcionar funciones de diagnóstico y otras funciones adicionales para el host, como MLD y *Neighbor Discovery*. Los mensajes informativos ICMPv6* incluyen Solicitud de eco (**Echo Request**) y Respuesta de eco (**Echo Reply**).

Encabezado de ICMPv6

Los campos del encabezado ICMPv6 son los siguientes:

Tipo (**Type**): indica el tipo de mensaje ICMPv6. El tamaño de este campo es de 8 bits. En los mensajes de error ICMPv6, el bit de orden superior se establece en el valor 0. En los mensajes informativos ICMPv6, el bit de orden superior se establece en el valor 1.

Código (**Code**): distingue entre varios mensajes dentro de un tipo de mensaje dado. El tamaño de este campo es de 8 bits. Si sólo hay un mensaje de un tipo dado, el campo "Code" se establece en 0.

Suma de comprobación (**Checksum**): almacena una suma de comprobación del mensaje ICMP. El tamaño de este campo es de 16 bits. El pseudo-encabezado de IPv6 se agrega al mensaje ICMPv6 cuando se calcula la suma de comprobación.

Cuerpo del mensaje (**Message body**): contiene datos específicos del mensaje ICMPv6.

Mensajes de error ICMPv6

Los mensajes de error ICMPv6 se utilizan para informar de errores de reenvío o entrega por parte de un ruteador o del host de destino.

---

* Los mensajes informativos se describen en RFC 2463

Destino inaccesible (Destination Unreachable)

El *ruteador* o el host de destino envía un mensaje ICMPv6 *Destination Unreachable* cuando el paquete no se puede reenviar a su destino.

Paquete demasiado grande (Packet Too Big)

Se envía un mensaje ICMPv6 *Packet Too Big* cuando el paquete no se puede reenviar debido a que la unidad MTU del vínculo de reenvío es menor que el tamaño del paquete IPv6.

El mensaje *Packet Too Big* se utiliza para el proceso Descubrimiento MTU de ruta de acceso de IPv6 (*Path MTU Discovery).*

Fin de tiempo de espera (Time Exceeded)

Normalmente, un ruteador envía un mensaje ICMPv6 *Time Exceeded* cuando el campo *Hop Limit* del encabezado de IPv6 es cero al recibir el paquete o después de reducir su valor durante el proceso de reenvío.

En el mensaje *Time Exceeded*, se establece cuando se sobrepasa el tiempo de reensamblado de la fragmentación del host de destino. La recepción de mensajes *Time Exceeded* indica que el límite de saltos de los paquetes salientes no es suficientemente grande para llegar al destino o que existe un bucle de ruteo.

Problema de parámetro (Parameter Problem)

El mensaje ICMPv6 *Parameter Problem* es enviado por un ruteador o por el destino. Ocurre cuando se detecta un error en el encabezado de IPv6 o en un encabezado de extensión, e impide que continúe el procesamiento de IPv6.

En el mensaje *Parameter Problem*, indica el desplazamiento en bytes del paquete IPv6 en el que se detectó el error. El valor del campo "Pointer" se establece en el desplazamiento correcto incluso cuando la ubicación del error no esté en la parte del paquete descartado.

Mensajes informativos ICMPv6

Los mensajes informativos ICMPv6, definidos en RFC 2463, proporcionan capacidades de diagnóstico para la solución de problemas.

Solicitud de eco (Echo Request)

El mensaje ICMPv6 *Echo Request* se envía a un destino para solicitar un mensaje *Echo Reply* de inmediato. El servicio de mensajes *Echo Request/Echo Reply* proporciona un diagnóstico simple para la solución de diversos problemas de posibilidad de acceso y enrutamiento.

En el mensaje *Echo Request*, los campos Identificador (*Identifier*) y número de secuencia (*Sequence Number*) se establecen mediante el host de envío y se utilizan para hacer coincidir un mensaje *Echo Reply* entrante con su mensaje *Echo Request* correspondiente. El campo Datos (*Data*) contiene datos opcionales y también lo establece el host de envío.

Respuesta de eco (Echo Reply)

Se envía un mensaje ICMPv6 *Echo Reply* en respuesta a la recepción de un mensaje ICMPv6 *Echo Request*.

En el mensaje *Echo Reply*, los campos Identifier, Sequence Number y Data se establecen con los mismos valores que los del mensaje *Echo Request* que solicitó inicialmente el mensaje *Echo Reply*.

Diferencias entre los mensajes ICMPv4 e ICMPv6

En la tabla 2.2 se muestran los mensajes ICMPv4 y sus equivalentes en ICMPv6 a manera de resumen.

Mensaje ICMPv4	Equivalente en ICMPv6
Destino inaccesible: red inaccesible (Destination Unreachable-Network unreachable)	Destino inaccesible: no hay ruta al destino (Destination Unreachable-No route to destination)
Destino inaccesible: host inaccesible (Destination Unreachable-Host unreachable)	Destino inaccesible: dirección inaccesible (Destination Unreachable-Address unreachable)
Destino inaccesible: protocolo inaccesible (Destination Unreachable-Protocol unreachable)	Problema de parámetro: no se reconoce el campo Next Header (Parameter Problem-Unrecognized Next Header field)
Destino inaccesible: puerto inaccesible (Destination Unreachable-Port unreachable)	Destino inaccesible: puerto inaccesible (Destination Unreachable-Port unreachable)
Destino inaccesible: se necesita fragmentación y DF (Destination Unreachable-Fragmentation needed and DF set)	Paquete demasiado grande (Packet Too Big)
Destino inaccesible: comunicación con el host de destino prohibida administrativamente (Destination Unreachable-Communication with destination host administratively prohibited)	Destino inaccesible: comunicación con el destino prohibida administrativamente (Destination Unreachable-Communication with destination administratively prohibited)

Fin de tiempo de espera caducó TTL (Time Exceeded-TTL expired)	Fin de tiempo de espera: se excedió el límite de saltos (Time Exceeded-Hop Limit exceeded)
Fin de tiempo de espera caducó el cronómetro de fragmentación (Time Exceeded-Fragmentation timer expired)	Fin de tiempo de espera: se excedió del cronómetro de fragmentación (Time Exceeded-Fragmentation timer exceeded)
Problema de parámetro (Parameter Problem)	Problema de parámetro (Parameter Problem)
Paquetes de control de flujo (Source Quench)	Este mensaje no está implementado en IPv6.
Redirección (Redirect)	Redirección para descubrimiento de vecino (Mensaje Neighbor Discovery Redirect)

Tabla 2.2 Mensajes ICMPv4 y sus equivalentes en ICMPv6

Descubrimiento de MTU de ruta de acceso

La unidad MTU de ruta de acceso es la MTU (de sus siglas en ingles, Maximum transmission Unit) de vínculo mínima de todos los vínculos que hay en una ruta de acceso entre un origen y un destino. Los paquetes IPv6 con un tamaño máximo de MTU de ruta de acceso no necesitan que el host los fragmente y todos los ruteadores de la ruta de acceso los reenviarán correctamente. Para descubrir la unidad MTU de ruta de acceso, el nodo de envío utiliza la recepción de mensajes ICMP Packet Too Big[18].

La unidad MTU de ruta de acceso se descubre mediante el siguiente proceso:

1. El nodo de envío asume que la unidad MTU de la ruta de acceso es la MTU de vínculo de la interfaz en la que se está reenviando el tráfico.

2. El nodo de envío envía datagramas IP con el tamaño de MTU de ruta de acceso.

3. Si un ruteador no puede reenviar el paquete a través de un vínculo con una MTU de vínculo menor que el tamaño del paquete, descarta el paquete IPv6 y devuelve un mensaje Packet Too Big al nodo de envío. El mensaje ICMP Packet Too Big contiene la unidad MTU del vínculo en el que se produjo el error de reenvío.

4. El nodo de envío configura la unidad MTU de ruta de acceso para los paquetes que se envían al destino con el valor del campo MTU en el mensaje ICMPv6 Packet Too Big.

[18] Miller Mark, Implementing IPv6, p. 129

El nodo de envío vuelve a empezar en el paso 2 y repite los pasos 2 a 4 tantas veces como sea necesario para descubrir la unidad MTU de ruta de acceso. La unidad MTU de ruta de acceso se determina cuando no se reciben mensajes ICMPv6 Packet Too Big adicionales o cuando se recibe un mensaje de confirmación del destino.

En el RFC 1981, se recomienda que los nodos IPv6 admitan el descubrimiento de MTU de ruta de acceso. Aquéllos que no lo hagan, deben utilizar la unidad MTU de vínculo mínima de 1.280 bytes como MTU de ruta de acceso.

Las disminuciones de MTU de ruta de acceso se descubren inmediatamente a través de la recepción de mensajes ICMP *Packet Too Big*. El nodo de envío debe detectar los incrementos en la MTU de ruta de acceso. El nodo de envío puede intentar enviar un paquete IPv6 mayor después de un mínimo de 5 minutos (se recomienda 10 minutos) al recibir un mensaje ICMPv6 *Packet Too Big*.

Descubrimiento de vecino

Descubrimiento de vecino (Neighbor Discovery, ND) de IPv6 es un conjunto de mensajes y procesos que determinan las relaciones entre nodos vecinos. ND reemplaza a los procesos ARP, ICMP Router Discovery (Descubrimiento de ruteadores) e ICMP Redirect (Redirección) que se utilizaban en IPv4 y proporciona funciones adicionales.

ND es utilizado por:

- Los hosts, para descubrir ruteadores vecinos.
- Los hosts, para descubrir direcciones, prefijos de direcciones y otros parámetros de configuración.
- Los nodos, para resolver la dirección de nivel de vínculo de un nodo vecino al que se va a reenviar un paquete IPv6 y determinar cuándo ha cambiado la dirección de nivel de vínculo de un nodo vecino.
- Los nodos, para determinar si aún se puede tener acceso a un vecino.
- Los ruteadores, para anunciar su presencia, los parámetros de configuración de host y los prefijos en el vínculo.
- Los ruteadores, para informar a los hosts de una dirección de salto siguiente mejor para el reenvío de paquetes a un destino específico.

En la tabla 2.3 se describen los procesos ND documentados en RFC 2461.

Proceso	Descripción
Descubrimiento de ruteadores	Proceso por el que un host descubre los ruteadores locales de un vínculo conectado. Equivale al proceso Descubrimiento de ruteador (Router Discovery) de ICMPv4.

Descubrimiento de prefijos	Proceso por el que los hosts descubren los prefijos de red para destinos de vínculos locales. Es similar al proceso Solicitud y respuesta de máscara de dirección (Address Mask Request/Reply) de ICMPv4.
Descubrimiento de parámetros	Proceso por el que los hosts descubren parámetros de funcionamiento adicionales, incluida la unidad MTU de vínculo y el límite de saltos predeterminado para los paquetes salientes.
Configuración automática de direcciones	Proceso que consiste en configurar direcciones IP para interfaces en presencia o en ausencia de un servidor de configuración de direcciones con estado, como la versión 6 del Protocolo de configuración dinámica de host (DHCPv6).
Resolución de direcciones	Proceso por el que los nodos resuelven la dirección IPv6 de un vecino en su dirección de nivel de vínculo. Equivale a ARP en IPv4.
Determinación del salto siguiente	Proceso por el que un nodo determina la dirección IPv6 del vecino al que se envía un paquete basándose en la dirección de destino. La dirección de reenvío o de salto siguiente es la dirección de destino o la dirección de un ruteador predeterminado en el vínculo.
Detección de inaccesibilidad a un vecino	Proceso por el que un nodo determina que el nivel IPv6 de un vecino ya no recibe paquetes.
Detección de dirección duplicada	Proceso por el que un nodo determina que un nodo vecino aún no utiliza una dirección considerada para el uso. Equivale a utilizar tramas ARP gratuitas en IPv4.
Función de redirección	Proceso que consiste en informar al host de una dirección IPv6 mejor para el primer salto para llegar a un destino. Equivale al mensaje ICMP Redirect (Redirección) de IPv4.

Tabla 2.3 Procesos de Neighbor Discovery (Descubrimiento de vecinos) en IPv6

Formato de los mensajes Descubrimiento de vecino (Neighbor Discovery)

Al igual que los mensajes Descubrimiento de escucha de multidifusión (Multicast Listener Discovery, MLD), los mensajes Neighbor Discovery (ND) utilizan la estructura de mensajes de ICMPv6. Los mensajes ND constan de un encabezado de mensaje ND, compuesto por un encabezado ICMPv6 y datos específicos del mensaje ND, además de cero o más opciones de ND, tal como se muestra en la figura 2.18.

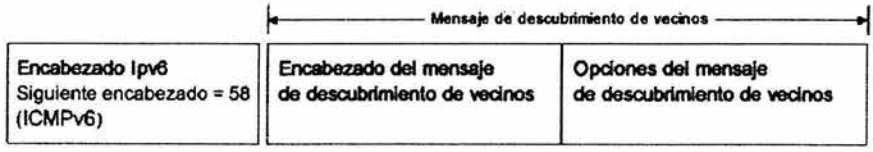


Figura 2.18 Formato de un mensaje Descubrimiento de vecino

Todas las funciones de Neighbor Discovery (ND) de IPv6 se realizan con cinco mensajes ND distintos:

- Solicitud de ruteador (**Router Solicitation**)
- Anuncio de ruteador (**Router Advertisement**)
- Solicitud de vecino (**Neighbor Solicitation**)
- Anuncio de vecino (**Neighbor Advertisement**)
- Redirección (**Redirect**)

En la figura 2.19 se ilustra las funciones de Neighbor Discovery

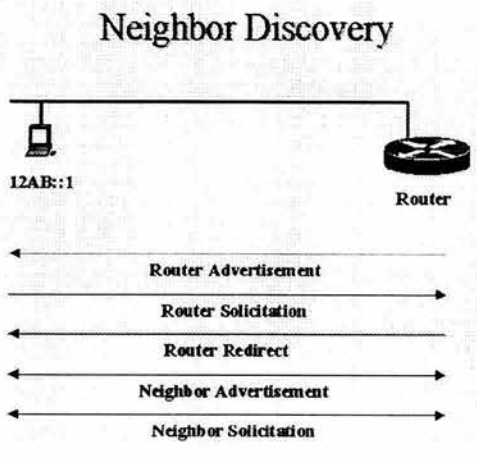


Figura 2. 19 Descubrimiento de vecino (Neighbor Discovery)

Solicitud de ruteador (Router Solicitation)

El mensaje *Router Solicitation* es enviado por los hosts IPv6 para descubrir los ruteadores IPv6 que hay en el vínculo. Un host envía una solicitud de ruteador de multidifusión para que los ruteadores IPv6 respondan inmediatamente, en vez de esperar un mensaje periódico Anuncio de ruteador *(Router Advertisement)*.

Por ejemplo, si el vínculo local es Ethernet, en el encabezado Ethernet del mensaje Router Solicitation:

- El campo Dirección de origen (Source Address) se establece en la dirección MAC del adaptador de red de envío.
- El campo Dirección de destino (Destination Address) se establece en el valor 33-33-00-00-00-02.

En el encabezado IPv6 del mensaje *Router Solicitation* hay los campos siguientes:

- El campo *Source Address* se establece en la dirección IPv6 asignada a la interfaz de envío o con la dirección IPv6 no especificada (::).
- El campo *Destination Address* se establece en la dirección de multidifusión local de vínculo de todos los ruteadores (FF02::2).
- El campo *Hop Limit* se establece en el valor 255.

Anuncio de ruteador (Router Advertisement)

Los ruteadores IPv6 envían el mensaje Router Advertisement periódicamente o en respuesta a la recepción de un mensaje Router Solicitation. Contiene la información que necesitan los hosts para determinar los prefijos de vínculo, la unidad MTU de vínculo, si se utiliza o no la configuración automática de direcciones y el tiempo durante el que las direcciones creadas mediante la configuración automática de direcciones son válidas y preferidas.

Por ejemplo, si el vínculo local es Ethernet, en el encabezado Ethernet del mensaje Router Advertisement:

- El campo Source Address se establece en la dirección MAC del adaptador de red de envío.
- El campo Destination Address se establece en 33-33-00-00-00-01 para un anuncio de enrutamiento periódico o la dirección MAC de unidifusión del host que envió una solicitud de ruteador.

En el encabezado IPv6 del mensaje *Router Advertisement*:

- El campo *Source Address* se establece en la dirección local de vínculo asignada a la interfaz de envío.

- El campo *Destination Address* se establece como dirección de multidifusión local de vínculo de todos los nodos (FF02::1) o la dirección IPv6 de unidifusión del host que envió el mensaje Router Solicitation.

- El campo *Hop Limit* se establece en el valor 255.

Solicitud de vecino (Neighbor Solicitation)

Los hosts IPv6 envían el mensaje *Neighbor Solicitation* para descubrir la dirección de nivel de vínculo de un nodo IPv6 en un vínculo. Incluye la dirección de nivel de vínculo del remitente. Las solicitudes de vecino típicas son de multidifusión para la resolución de direcciones y de unidifusión cuando se está comprobando la posibilidad de acceso a un nodo vecino.

Por ejemplo, si el vínculo local es Ethernet, en el encabezado Ethernet del mensaje Neighbor Solicitation:

- El campo *Source Address* se establece en la dirección MAC del adaptador de red de envío.

- Para una solicitud de vecino multidifusión, el campo *Destination Address* se establece en la dirección MAC Ethernet que corresponde a la dirección IP de multidifusión del nodo solicitado del destino. Para una solicitud de vecino unidifusión, el campo *Destination Address* se establece en la dirección MAC de unidifusión del vecino.

En el encabezado IPv6 del mensaje *Neighbor Solicitation*:

- El campo *Source Address* se establece en la dirección IPv6 asignada a la interfaz de envío o, durante la detección de detecciones duplicadas, con la dirección IPv6 no especificada (::).

- Para una solicitud de vecino multidifusión, el campo *Destination Address* se establece en la dirección de multidifusión de nodo solicitado del destino. Para una solicitud de vecino unidifusión, el campo *Destination Address* se establece en la dirección IP de unidifusión del destino.

- El campo *Hop Limit* se establece en el valor 255.

Anuncio de vecino (Neighbor Advertisement)

Un nodo IPv6 envía el mensaje *Neighbor Advertisement* en respuesta a la recepción de un mensaje *Neighbor Solicitation*. Un nodo IPv6 también envía anuncios de vecino no solicitados para informar a los nodos vecinos de los cambios en las direcciones de nivel de vínculo. El mensaje *Neighbor Advertisement* contiene información que necesitan los nodos para determinar el tipo de mensaje *Neighbor Advertisement*, la dirección de nivel de vínculo del remitente y la función del remitente en la red.

Por ejemplo, si el vínculo local es Ethernet, en el encabezado Ethernet del mensaje *Neighbor Advertisement*:

- El campo *Source Address* se establece en la dirección MAC del adaptador de red de envío.

- Para el anuncio de vecino solicitado, el campo *Destination Address* se establece en la dirección MAC de unidifusión del remitente de la solicitud de vecino inicial. Para un anuncio de vecino no solicitado, el campo *Destination Address* se establece en 33-33-00-00-00-01, que es la dirección MAC Ethernet correspondiente a la dirección de multidifusión local de vínculo de todos los nodos.

En el encabezado IPv6 del mensaje *Neighbor Advertisement*:

- El campo *Source Address* se establece en la dirección local de vínculo asignada a la interfaz de envío.

- Para un anuncio de vecino solicitado, el campo *Destination Address* se establece en la dirección IP de unidifusión del remitente de la solicitud de vecino inicial. Para un anuncio de vecino no solicitado, el *campo Destination Address* se establece en la dirección de multidifusión local de vínculo de todos los nodos (FF02::1).

- El campo *Hop Limit* se establece en el valor 255.

Redirección (Redirect)

Un ruteador de IPv6 envía el mensaje *Redirect* para informar a un host de origen de la existencia de una dirección mejor para el primer salto a un destino determinado. Los mensajes *Redirect* sólo son enviados por los ruteadores de tráfico de unidifusión, son sólo de unidifusión para los hosts de origen y únicamente son procesados por hosts.

Por ejemplo, si el vínculo local es Ethernet, en el encabezado Ethernet del mensaje *Redirect*:

- El campo *Source Address* se establece en la dirección MAC del adaptador de red de envío.

- El campo *Destination Address* se establece en la dirección MAC de unidifusión del remitente de origen.

En el encabezado IPv6 del mensaje *Neighbor Advertisement*:

- El campo *Source Address* se establece en la dirección local de vínculo asignada a la interfaz de envío.

- El campo *Destination Address* se establece en la dirección IP de unidifusión del host de origen.

- El campo *Hop Limit* se establece en el valor 255.

Ejemplos de *Neighbor Solicitation* y *Neighbor Advertisement* respectivamente, capturados en éste proceso por medio de un *sniffer*.

IEEE 802.3/Ethernet DIX V2 Header	ICMPv6 - Internet Control Message Protocol (Version 6)
Decode Status : - Frame Length : 86 Destination Address : 33-33-FF-F3-6D-41 Source Address : 00-60-08-BE-68-2C, JANUS Frame Format : Ethernet DIX V2 Ethertype : 0x86DD (IPv6) Frame Checksum : Good IP - Internet Protocol (Version 6) Decode Status : - Version Number : 6 (IP Version 6) Traffic Class : 0x 00, Flow Label : 0x0 Payload Length : 32 Next Header : 58 (ICMPv6) Hop Limit : 255 Source Address : FE80:0:0:0:260:8FF:FEBE:682C Destination Address : FF02:0:00:0:1:FFF3:6D41	Decode Status : - ICMP Type : 135 (Neighbor Solicitation) Neighbor Solicitation Code : 0 Checksum : 41184 Target Address : 3FFE:8070:1:0:2D0:58FF:FEF3:6D41 Options Option : 1 (Source Link Layer Address) Option Length : 1 Source Link Layer Address : 00-60-08-BE-68-2C, JANUS

Figura 2.20 Neighbor solicitation

IEEE 802.3/EthernetDIX V2 Header	ICMPv6 -Internet Control Message Protocol (Version 6)
Decode Status : -	Decode Status : -
Frame Length : 86	ICMP Type : 136 (Neighbor Advertisement)
Destination Address : 00-60-08-BE-68-2C, JANUS	Neighbor Advertisement Code : 0
Source Address : 00-D0-58-F3-6D-41	Checksum : 3801
Frame Format : Ethernet DIX V2	Flags : 0xE0
Ethertype : 0x86DD (IPv6)	1 = Sender is a Router
Frame Checksum : Good	1 = Response to a Neighbor Solicitation
IP - Internet Protocol (Version 6)	1 = Override Link-Layer Entry
Decode Status : -	Target Address : 3FFE:8070:1:0:2D0:58FF:FEF3:6D41
Version Number : 6 (IP Version 6)	Options
Traffic Class : 0x70	Option : 2 (Target Link Layer Address)
Flow Label : 0x0	Option Length : 1
Payload Length : 32	Target Link Layer Address : 00-D0-58-F3-6D-41
Next Header : 58 (ICMPv6)	
Hop Limit : 255	
Source Address : FE80:0:0:0:2D0:58FF:FEF3:6D41	
Destination Address : FE80:0:0:0:260:8FF:FEBE:682C	

Figura 2.21 Neighbor advertisement

Opciones de Descubrimiento de vecino

Las opciones de los mensajes ND proporcionan información adicional, que normalmente indica direcciones MAC, prefijos de red en el vínculo, información de MTU en el vínculo y datos de redirección.

Para asegurarse de que los mensajes ND recibidos se originaron en un nodo del vínculo local, todos los mensajes ND se envían con un límite de saltos de 255. Cuando se recibe un mensaje ND, se comprueba el campo Límite de saltos del encabezado IPv6. Si no se establece en el valor 255, el mensaje se descarta sin notificarlo. La comprobación de que un mensaje ND tiene un límite de saltos de 255 proporciona protección ante ataques en la red basados en ND desde nodos situados fuera del vínculo. Con un límite de saltos de 255, un ruteador no podría reenviar el mensaje ND desde un nodo situado fuera del vínculo.

Las opciones de *Neighbor Discovery* tienen el formato Tipo-*Longitud*-Valor. El campo Tipo *(Type)* de 8 bits indica el tipo de opción de ND.

Opción Dirección de nivel de vínculo de origen y destino (Source/Target Link-Layer Address)

La opción *Source Link-Layer Address* indica la dirección de nivel de vínculo del remitente del mensaje ND. La opción *Source Link-Layer Address* se incluye en los mensajes *Neighbor Solicitation*, *Router Solicitation* y *Router Advertisement*. La opción *Source Link-Layer Address* no se incluye cuando la dirección de origen del mensaje ND es la dirección no especificada (::).

La opción *Target Link-Layer Address* indica la dirección de nivel de vínculo del nodo vecino al que se deben dirigir los paquetes IPv6. La opción *Target Link-Layer Address* se incluye en los mensajes *Neighbor Advertisement* y *Redirect* (Redirección).

Opción Información de prefijo (Prefix Information)

La opción *Prefix Information* se envía en mensajes *Router Advertisement* para indicar los prefijos de las direcciones e información acerca de la configuración automática de direcciones. En un mensaje *Router Advertisement*, puede haber varias opciones *Prefix Information*, que indican varios prefijos de direcciones.

Opción Encabezado de redirección (Redirected Header)

La opción *Redirected Header* se envía a los mensajes *Redirect* para especificar el paquete IPv6 que hizo que el ruteador enviara un mensaje *Redirect*. Puede contener todo el paquete IPv6 redirigido o una parte, según el tamaño del paquete IPv6 que se envió inicialmente.

Opción MTU

La opción MTU se envía en mensajes *Router Advertisement* para indicar la unidad MTU de IPv6 del vínculo. Normalmente, esta opción sólo se utiliza cuando la MTU de IPv6 para un vínculo no es bien conocida o tiene que establecerse debido a una configuración de puente de transacciones. La opción MTU suplanta a la unidad MTU de IPv6 de la que informa el hardware de interfaz.

Las diferencias en las unidades MTU de IPv6 entre nodos de la misma red no se detectan a través del proceso **Descubrimiento de MTU de ruta de acceso** (Path MTU Discovery). La opción de MTU se utiliza para indicar la unidad MTU de IPv6 de nivel superior que admiten todas las tecnologías de nivel de vínculo en el segmento de red, ver figura 2.22

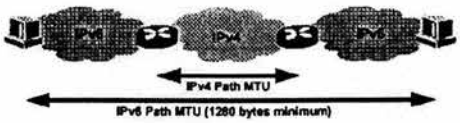


Figura 2.22 Path MTU Discovery

Procesos de Descubrimiento de vecino (Neighbor Discovery)

El protocolo Neighbor Discovery (ND) proporciona intercambios de mensajes para los siguientes procesos:

- Resolución de direcciones (incluida la detección de direcciones duplicadas)
- Descubrimiento de ruteadores (incluye descubrimiento de parámetros y prefijos)
- Detección de inaccesibilidad a un vecino
- Función de redirección

Para facilitar interacciones entre nodos vecinos, en RFC 2461 se definen las siguientes estructuras de datos de host, se da un ejemplo de cómo se puede almacenar información para procesos ND:

- Caché de vecino

  Almacena la dirección IP de vínculo de un vecino, su dirección de nivel de vínculo correspondiente y una indicación de la posibilidad de acceso al vecino. La caché de vecino equivale a la caché de ARP en IPv4.

- Caché de destino

  Almacena información acerca de direcciones IP de salto siguiente o de reenvío para los destinos a los que se ha enviado tráfico recientemente. Las entradas de la caché de destino contienen la dirección IP de destino (local o remota), la dirección IP de salto siguiente resuelta anteriormente y la unidad MTU de ruta de acceso para el destino.

- Lista de prefijos

  Enumera los prefijos del vínculo. Cada entrada de la lista de prefijos define un intervalo de direcciones IP para destinos a los que se puede tener acceso directo (vecinos). Esta lista se llena con prefijos anunciados por ruteadores en el mensaje *Router Advertisement* (Anuncio de ruteador).

- Lista de ruteadores predeterminados

Enumera direcciones IP que corresponden a ruteadores del vínculo que envían mensajes *Router Advertisement* y pueden ser ruteadores predeterminados.

Resolución de direcciones

El proceso de resolución de direcciones para los nodos IPv6 consiste en el intercambio de mensajes *Neighbor Solicitation* y *Neighbor Advertisement* para resolver la dirección de nivel de vínculo de la dirección de salto siguiente en el vínculo para un destino dado. El host remitente envía un mensaje *Neighbor Solicitation* de multidifusión para la interfaz apropiada. La dirección de multidifusión del mensaje *Neighbor Solicitation* es la dirección de multidifusión de nodo solicitado derivada de la dirección IP de destino. El mensaje *Neighbor Solicitation* incluye la dirección de nivel de vínculo del host de envío en la opción *Source Link-Layer Address*.

Cuando el host de destino recibe el mensaje *Neighbor Solicitation*, actualiza su propia caché de vecino basándose en la dirección de origen del *mensaje Neighbor Solicitation* y la dirección de nivel de vínculo especificada en la *opción Source Link-Layer Address*. A continuación, el nodo de destino envía un anuncio de vecino de unidifusión al remitente del mensaje *Neighbor Solicitation*. El mensaje *Neighbor Advertisement* incluye la opción *Target Link-Layer Address*.

Después de recibir el mensaje *Neighbor Advertisement* del destino, el host de envío actualiza su caché de vecino con una entrada para el destino basada en la información que se especifique en la opción *Target Link-Layer Address*. En este momento, se puede enviar tráfico IPv6 de unidifusión entre el host de envío y el destino del mensaje *Neighbor Solicitation*.

2.5 Autoconfiguración

La Autoconfiguración IPv6 está pensada para ser transparente para la mayoría de los usuarios y soportar más facilidades que IPv4

Uno de los aspectos más útiles de IPv6 es su capacidad para configurarse automáticamente, incluso sin ayuda de un protocolo de configuración con estado como el Protocolo de Configuración Dinámica de Host para IPv6 (Dinamic Host Configuration Protocol, DHCPv6). De forma predeterminada, un host IPv6 puede configurar una dirección local de enlace para cada interfaz. Mediante el proceso de descubrimiento de ruteadores, un host también puede determinar las direcciones de los ruteadores, otros parámetros de configuración, direcciones adicionales y prefijos en el vínculo. En el mensaje Anuncio de ruteador (Router Advertisement) incluye una indicación de si debe utilizarse un protocolo de configuración de direcciones con estado.

La configuración automática de direcciones sólo se puede llevar a cabo con interfaces compatibles con la multidifusión.

Estados de direcciones configuradas automáticamente

Las direcciones que se configuran automáticamente se encuentran en uno o varios de los estados siguientes:

- Provisional (**Tentative**)

  Se está comprobando si la dirección es única. La comprobación se realiza mediante el proceso de detección de direcciones duplicadas. Un nodo no puede recibir tráfico de unidifusión para una dirección provisional. Sin embargo, puede recibir y procesar mensajes Anuncio de vecino (*Neighbor Advertisement*) de multidifusión enviados como respuesta al mensaje Solicitud de vecino (*Neighbor Solicitation*) que se envió durante el proceso de detección de direcciones duplicadas.

- Preferida (**Preferred**)

  Dirección cuya unicidad se ha comprobado. Un nodo puede enviar y recibir tráfico de unidifusión de direcciones preferidas. El período de tiempo que una dirección puede mantenerse en estado de preferencia está determinado por el campo de Tiempo de vida preferido (*Preferred Lifetime*) en la opción Información de prefijo (*Prefix Information*) de un mensaje Anuncio de ruteador *(Router Advertisement*).

- Desaprobada (**Deprecated**)

  Dirección que, aunque es válida, no es recomendable utilizar para una nueva comunicación. En las sesiones de comunicación ya existentes aún pueden utilizarse direcciones desaprobadas. Un nodo puede enviar y recibir tráfico de unidifusión a y de direcciones desaprobadas.

- Válida (**Valid**)

  Dirección desde la que se puede enviar y recibir tráfico de unidifusión. El estado de dirección válida incluye los estados de dirección preferida y desaprobada. El tiempo que una dirección se mantiene en estado de validez está determinado por el campo Tiempo de vida válido (*Valid Lifetime)* en la opción *Prefix Information* de un mensaje *Router Advertisement*. El tiempo de vida válido debe ser igual o mayor que el tiempo de vida preferido.

- No válida (**Invalid**)

Dirección para la que un nodo ya no puede enviar o recibir tráfico de unidifusión. Una dirección pasa al estado de no válida cuando caduca el tiempo de vida válido.

En la figura 2.23 se muestra la relación entre los estados de una dirección configurada automáticamente y los tiempos de vida preferido y válido.

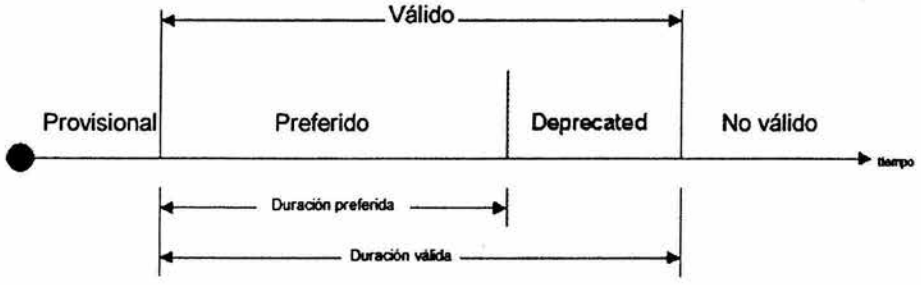


Figura 2.23 Estado y tiempos de vida de una dirección configurada automáticamente

Con excepción de una configuración automática para direcciones de enlace locales, la configuración automática de direcciones sólo se especifica para los hosts. Los ruteadores deben obtener los parámetros de configuración y de dirección por otros medios, tales como la configuración manual.

2.5.1 Tipos de configuración automática

Hay tres tipos de perfiles de configuración automática:

1. Sin estado (**Stateless**)

   La configuración de direcciones se basa en la recepción de mensajes Anuncio de enrutador *(Router Advertisement)* con los indicadores Configuración de direcciones administradas (*Managed Address Configuration*) y otras configuraciones con estado (*Other Stateful Configuration*) establecidos en el valor 0, y una o varias opciones de Información de prefijo (*Prefix Information*).

   La configuración *Stateless* ocurre cuando el ruteador configura al host con una dirección IPv6. El Ruteador inicializa a los hosts que se quieren unir a la red con la dirección multicast FE02::1 y permite al nodo usar su dirección de red. Esta será probablemente su dirección física de red (MAC) de 48-bits. Por tanto, para la autoconfiguración *stateless* no se requiere la presencia de servidores

Si un host se une a la red este envía una solicitud a todos los routers usando una dirección Multicast del tipo FF02::2. El router le ayudará a determinar el prefijo y otros parámetros de ruteo.

2. Con estado (**stateful**)

   La Autoconfiguración *Stateful* consiste en implementar la funcionalidad de DHCP para IPv6 (DHCPv6), para obtener direcciones y otras opciones de configuración. Los hosts son configurados con una dirección y otros parámetros por un servidor de DHCPv6. Un host utiliza la configuración de direcciones con estado cuando recibe mensajes *Router Advertisement* sin opciones de prefijo en los que el indicador *Managed Address Configuration* o el indicador *Other Stateful Configuration* están establecidos en el valor 1.

   Los servidores mantienen, por tanto, una base de datos con todas las direcciones que han sido asignadas y a qué hosts, al igual que todo lo relacionado con el resto de los parámetros.

3. Ambos

   La configuración se basa en la recepción de mensajes *Router Advertisement* con opciones *Prefix Information* y el indicador *Managed Address Configuration* o el indicador *Other Stateful Configuration* establecidos en el valor 1.

Para todos los tipos, se configura siempre una dirección local de vínculo.

2.5.2 Proceso de autoconfiguración

El proceso de autoconfiguración para un nodo IPv6 es el siguiente:

1. Se deriva una dirección local de vínculo provisional a partir del prefijo local de vínculo FE80::/64 y el identificador de interfaz de 64 bits.

2. Mediante el proceso de detección de direcciones duplicadas, para comprobar la unicidad de una dirección local de vínculo provisional, se envía un mensaje *Neighbor Solicitation* con el campo de Dirección de destino (*Target Address*) establecido en la dirección local de enlace provisional.

3. Si se envía un mensaje *Neighbor Advertisement* en respuesta al mensaje *Neighbor Solicitation* que se recibió, esto indica que otro nodo del vínculo local utiliza la dirección local de vínculo provisional y se detiene la configuración automática de direcciones. En este momento, se debe realizar una configuración manual en el nodo.

4. Si no se recibe ningún mensaje *Neighbor Advertisement* (que se envía en respuesta al mensaje *Neighbor Solicitation*), se asume que la dirección local de enlace provisional es única y válida. Se inicializa la dirección local de enlace para la interfaz. La dirección de nivel de enlace de multidifusión de nodo solicitado correspondiente se registra con el adaptador de red.

Para un host IPv6, la configuración automática de direcciones continúa como se describe a continuación:

1. El host envía un mensaje Solicitud de Ruteador (*Router Solicitation*).

2. Si no se recibe ningún mensaje *Router Advertisement*, el host utiliza un protocolo de configuración de direcciones con estado para obtener direcciones y otros parámetros de configuración.

3. Si se recibe un *mensaje Router Advertisement*, se configuran los campos Límite de saltos (*Hop Limit*), Tiempo accesible (*Reachable Time*), Cronómetro de retransmisión (*Retrans Timer*) y MTU.

4. Para cada opción *Prefix Information* que se utilice:

- Si el indicador Autónomo (*Autonomous*) se establece en el valor 1, el prefijo y el identificador de interfaz de 64 bits se utilizan para obtener una dirección provisional derivada.

- El proceso de detección de direcciones duplicadas se utiliza para comprobar la unicidad de la dirección provisional.

- Si se utiliza la dirección provisional, no se inicializa el uso de la dirección para la interfaz.

- Si no se utiliza la dirección provisional, se inicializa la dirección. Este proceso incluye la configuración de los tiempos de vida de validez y preferido, basados en los campos Tiempo de vida válido (*Valid Lifetim*) y Tiempo de vida preferido (*Preferred Lifetime*) de la opción *Prefix Information*. También incluye el registro de la dirección de nivel de vínculo de multidifusión de nodo solicitado correspondiente con el adaptador de red.

5. Si el indicador Configuración de dirección administrada (Managed Address Configuration) del mensaje *Router Advertisement* está establecido en el valor 1, se utiliza un protocolo de configuración de direcciones con estado para obtener direcciones adicionales.

6. Si el indicador Otras configuraciones con estado (*Other Stateful Configuration)* del mensaje *Router Advertisement* está establecido en el

valor 1, se utiliza un protocolo de configuración de direcciones con estado para obtener parámetros de configuración adicionales.

2.6 Mecanismos de seguridad

Una de las grandes ventajas de IPv6 es, sin duda, la total integración de los mecanismos de seguridad, autenticación y confidencialidad (encriptación), dentro del núcleo del protocolo.

Se trata por tanto de algo obligatorio, y no adicionar ni "añadido" como en IPv4. Para ello, la siguiente cabecera puede tiene valores AH (*Authentication*) y ESP (*Encapsulation security Payload*), que permiten, básicamente, emplear las mismas extensiones de protocolo empleadas en IPv4, y que de hecho, al haber sido desarrolladas con posterioridad al inicio de los trabajos de IPv6, ya lo contemplan.

2.6.1 Autenticación

La autenticación es un mecanismo que proporciona valga la redundancia, autenticación e integridad. Esto no incluye confidencialidad, ya que los datagramas de IPv6 no están encriptados. IPv6 soporta diferentes algoritmos y técnicas de autenticación. El algoritmo MD5 es el estándar propuesto para ayudar a prevenir los problemas de compatibilidad dentro de Internet. Este mecanismo ayuda a eliminar gran cantidad de ataques a la red[19].

2.6.2 Encriptación

El segundo punto relativo a la seguridad en IPv6 proporciona integridad y confidencialidad a los datagramas IPv6. Como la autenticación, el encabezado de encapsulamiento también se cifra independientemente. Para conseguir interoperabilidad dentro de Internet, el algoritmo DES CBC está siendo usado como un estándar. El ESP usa el modo Túnel en donde se encripta todo el paquete IP y donde se localiza el nuevo encabezado sin encriptar.[20]

Sin embargo, la norma IPsec está incluida en el protocolo IPv6. Los elementos relacionados con IPv6 y asociados a la Seguridad son:

- IPsec: autenticación de dispositivos y cifrado de flujos.
- Infraestructuras asociadas: PKI, DNSsec, por mencionar algunas.

IPsec es solo un componente necesario para la seguridad en Internet

[19] Miller Mark, Implementing IPv6, p.231
[20] Ibidem p.233

2.7 Sistema de Nombres de Dominio versión 6 (DNSv6)

En RFC 1886 se describen varias mejoras realizadas en el Sistema de nombres de dominio (DNS) para IPv6, las cuales incluyen las novedades siguientes:

- Registro de recursos de direcciones de host (AAAA).
- Dominio IP6.INT para consultas inversas

Registro de recursos de direcciones de host (AAAA)

Se utiliza un nuevo tipo de registro de recursos DNS, AAAA (denominado "cuatro as", para resolver un nombre de dominio completo en una dirección IPv6. Es comparable al registro de recursos de direcciones de host (A) que se utiliza con IPv4. El tipo de registro de recursos se denomina AAAA (valor de tipo 128) porque las direcciones IPv6 de 128 bits son cuatro veces mayores que las direcciones IPv4 de 32 bits[21]. A continuación, se muestra un ejemplo de un registro de recursos AAAA:

host1.cuautitlan.unam.mx IN AAAA FEC0::2AA:FF:FE3F:2A1C

Un host debe especificar una consulta AAAA o una consulta general para un nombre de host específico para recibir datos de resolución de direcciones IPv6 en las secciones de respuesta de las consultas DNS.

El dominio IP6.INT

El dominio IP6.INT se ha creado para las consultas IPv6 inversas. Las consultas inversas, también denominadas consultas de puntero, determinan un nombre de host basado en la dirección IP. Para crear el espacio de nombres para las consultas inversas, cada dígito hexadecimal de la dirección IPv6 de 32 dígitos completamente expresada se convierte en un nivel independiente en el orden opuesto en la jerarquía de dominios inversa.

Por ejemplo, el nombre de dominio de búsqueda inversa para la dirección FEC0::2AA:FF:FE3F:2A1C (que de forma completa se expresa como FEC0:0000:0000:0000:02AA: 00FF:FE3F:2A1C) es:

C.1.A.2.F.3.E.F.F.F.0.0.A.A.2.0.0.0.0.0.0.0.0.0.0.0.0.0.C.E.F.IP6.INT.

La compatibilidad con DNS que se describe en RFC 1886 representa un método sencillo de asignar nombres de hosts a direcciones IPv6 y proporcionar una resolución de nombres inversa. Sin embargo, esta compatibilidad no proporciona un método sencillo para propagar los cambios a registros AAAA, debido a la nueva numeración del sitio o a la delegación de zonas de búsqueda inversa en límites de bits arbitrarios (IP6.INT se designa en límites de cuarteto). Estas cuestiones se

[21] Hagen Silvia, IPv6 Essentials, p.216

resuelven mediante un nuevo registro de recursos "A6" que se describe en el borrador de Internet titulado (Extensiones DNS que admiten cambiar la numeración y agregar direcciones IPv6" (*DNS Extensions to Support IPv6 Address Aggregation and Renumbering*).

Direcciones IPv4 y sus equivalentes en IPv6

En la tabla 2.3 se muestran direcciones y conceptos de direccionamiento de IPv4 y sus equivalentes en IPv6.

Dirección IPv4	**Dirección IPv6**
Clases de direcciones de Internet	No se ha implementado en IPv6
Direcciones de multidifusión (224.0.0.0/4)	Direcciones de multidifusión IPv6 (FF00::/8)
Direcciones de difusión	No se ha implementado en IPv6
La dirección no especificada es 0.0.0.0	La dirección no especificada es ::
La dirección de bucle de retroceso es 127.0.0.1	La dirección de bucle de retroceso es ::1
Direcciones IP públicas	Direcciones de unidifusión global agregables
Direcciones IP privadas (10.0.0.0/8, 172.16.0.0/12 y 192.168.0.0/16)	Direcciones locales de sitio (FEC0::/48)
Direcciones configuradas automáticamente (169.254.0.0/16)	Direcciones locales de vínculo (FE80::/64)
Representación de texto: notación decimal con puntos	Representación de texto: formato hexadecimal con signos de dos puntos, supresión de ceros a la izquierda y compresión de ceros. Las direcciones compatibles con IPv4 se expresan en notación decimal con puntos.
Representación de bits de red: máscara de subred en notación decimal o longitud de prefijo	Representación de bits de red: sólo longitud de prefijo
Resolución de nombres DNS: registro de recursos de direcciones de host IPv4 (A)	Resolución de nombres DNS: registro de recursos de direcciones de host IPv6 (AAAA)
Resolución de DNS inversa: dominio IN-ADDR.ARPA	Resolución de DNS inversa: dominio IP6.INT

Tabla 2.3 Asignación actual del espacio de direcciones de IPv6

2.8 Transición de IPv4 a IPv6

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

2.8.1 Mecanismos de transición

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

2.8.1.1 Doble pila (Dual-Stack)

El camino más lógico y evidente de transición es el uso simultáneo de ambos protocolos, en pilas separadas. Los dispositivos con ambos protocolos también se denominan "nodos IPv6/IPV4" [22].

De esta forma, un dispositivo con ambas pilas pueden recibir y enviar tráfico a nodos que sólo soportan uno de los dos protocolos (nodos sólo IPv4 o sólo IPv6).

El dispositivo tendrá una dirección en cada pila. Se pueden utilizar direcciones IPv4 e IPv6 relacionadas o no, y se pueden utilizar mecanismos manuales o automáticos para la asignación de las direcciones. El DNS podrá devolver la dirección IPv4, la dirección IPv6 o ambas, como se muestra en la figura 2.24

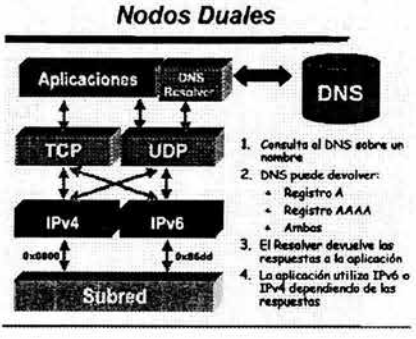


Figura 2.24 Nodos Duales

2.8.1.2 Túneles

Un túnel es una conexión virtual punto a punto en donde se encapsulan los paquetes IPv6 en los de IPv4 para transportarse por redes de IPv4, o sea que los

[22] Hagen silvia, Ipv6 Essentials, p.227, 228

túneles proporcionan un mecanismo para utilizar las infraestructuras IPv4 mientras la red IPv6 esta siendo implantada. Para configurar un túnel se necesitan dos direcciones IPv4 válidas globalmente y dos equipos que soporte IPv6.

Los túneles se clasifican según el mecanismo por el que el nodo que realiza el encapsulado determina la dirección del nodo extremo del túnel. Existen dos tipos de túneles configurados: manual y automáticamente, entre los automáticos están los siguientes: 6to4, 6over4, "tunnel Brokers", ISATAP (*Intra-Site Automatic Tunnel Addressing Protocol*), Tunneling IPv6 over UDP through NATs (TEREDO).

Los extremos finales del túnel siempre son los responsables de realizar la operación de encapsulado del paquete/es IPv6 en IPv4

Estos túneles pueden ser utilizados de formas diferentes:

- **Router a router**. Routers con doble pila (IPv6/IPv4) se conectan mediante una infraestructura IPv4 y transmiten tráfico IPv6. El túnel comprende un segmento que incluye la ruta completa, extremo a extremo, que siguen los paquetes IPv6

- **Host a router**. Hosts con doble pila se conectan a un router intermedio (también con doble pila), alcanzable mediante una infraestructura IPv4. El túnel comprende el primer segmento de la ruta seguida por los paquetes.

- **Host a host**. Hosts con doble pila interconectados para una infraestructura IPv4. El túnel comprende la ruta completa que siguen los paquetes.

- **Router a host**. Routers con doble pila que se conectan a hosts también con doble pila. El túnel comprende el último segmento de la ruta. Ver figura 2.25

En los dos primeros casos (router a router y host a router), el paquete IPv6 es "tunelizado" a un router. El extremo final de este tipo de túnel es un router intermedio que debe desencapsular el paquete IPv6 y reenviarlo a su destino final. En este caso, el extremo final del túnel es distinto del destino final del paquete por lo que la dirección en el paquete IPv6 no proporciona la dirección IPv4 del extremo final del túnel. La dirección del extremo final del túnel ha de ser determinada a través de información de configuración en el nodo que realiza el túnel. Es lo que se denomina "túnel configurado"[23].

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

[23] Miller Mark, Implementing IPv6, p.288, 290

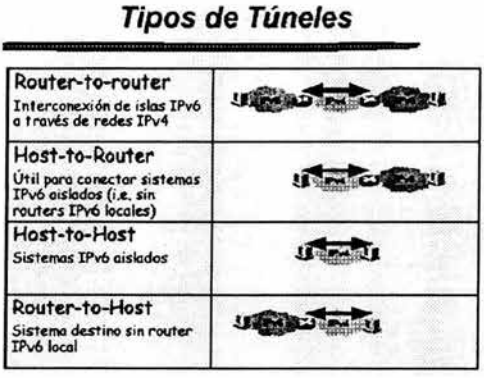


Figura 2.25 Tipos de túneles

6over4

Este mecanismo permite a hosts IPv6 aislados, sin conexión directa a routers IPv6, ser totalmente funcionales como dispositivos IPv6.

Para ello se emplean dominios IPv4 que soportan multicast como su enlace local virtual. Es decir, usamos multicast IPv4 como su "Ethernet virtual". De esta forma, estos hosts IPv6 no requieren direcciones IPv4 compatibles, ni túneles configurados[24].

Los extremos finales del túnel se determinan mediante ND. Es imprescindible que la subred IPv4 soporte multicast.

En la siguiente figura se da un ejemplo de este mecanismo usado en la UNAM

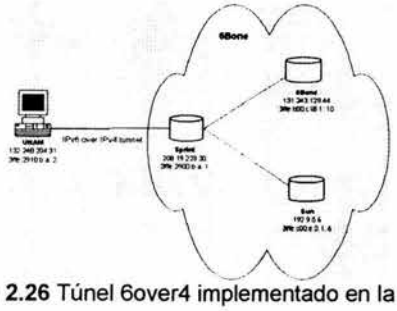


Figura 2.26 Túnel 6over4 implementado en la UNAM

[24] Palet Jordi, Tutorial de IPv6, p. 42, 43

6to4

En un método para conectar sitios IPv6 sobre la infraestructura IPv4 existente en Internet usando un prefijo de dirección especial(2002::/16) para proporcionar a sitios IPv6 aislados su propio espacio de direcciones IPv6. 6to4 es como un pseudo-ISP que proporciona conectividad en IPv6[27]; se puede usar para comunicarse directamente con otros sitios 6to4 y también con sitios de la red 6Bone.

El principal requisito para usar 6to4 es que se cuente con una dirección IPv4 global ruteable para el sitio.

ISATAP

Es un túnel de mecanismo automático que es diseñado para transportar paquetes IPv6 dentro de un sitio, no entre sitios.

ISATAP utiliza dual-stack para descubrir automaticamente routes y tuneles IPv6 sobre una infraestructura IPv4 dentro de un sitio. Cada hosts hace una petición a un router ISATAP dentro de un sitio.

Los paquetes que son enviados a Internet son ruteados vía ISATAP y paquetes destinados a otros hosts dentro de una mismo sitio son "tunelizados" directamente a el destino.

Los primeros 64 bits siguen el formato de direcciones unidifusión globlales agregables(enlace local, sitio local). Las organizaciones IANA y el IEEE OUI (Organizationally Unique Identifier) asignaron un identificador 00-00-5E y el tipo FE indicando que se trata de una dirección IPv4[28]. Los últimos 32 bits contienen una dirección IPv4 que puede ser escrito en formato decimal o hexadecimal, en la figura 2.27 se muestra un ejemplo de cómo se forma el identificador.

Por ejemplo si tenemos la siguiente dirección IPv6: 2001:0DB8:1234:5678::/64 y una dirección IPv4 140.173.129.8 ésta la convertimos a hexadecimal que nos quedaría 8CAD:8108. Así tenemos que:

2001:0DB8:1234:5678:0000:5EFE:8CAD:8108

otro ejemplo, si un host tiene asignado el prefijo 3FFE:1a05:510:200::/64, con una dirección IPv4 132.168.0.1 y representando esta dirección en hexadecimal tenemos :

3FFE:1a05:510:200:0:5EFE:84a8:1

[27] Hagen Silvia, IPv6 Essentials, p. 238-39
[28] Ibídem p. 241-242, 266

la dirección enlace local quedaría:

FE80::5EFE:132.168.0.1

Y la dirección sitio local:

FEC0::200:0:5EFE:132.168.0.1

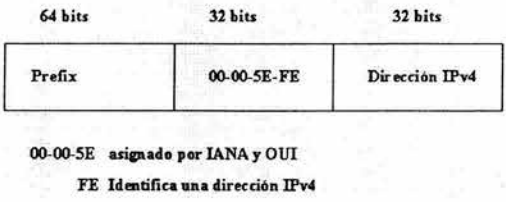


Figura 2.27 Estructura de ISATAP

Tunnel server y Tunnel Broker

Estos mecanismos se hacen indispensables para labores de investigación, dado que se requieren direcciones IPv6 y DNS permanentes. La diferencia con el mecanismo 6to4 es que el tunnel Broker no requiere la configuración de un router, en la figura 2.28 se pone un ejemplo de este mecanismo.

Se trata de ISP´s IPv6 "virtuales", proporcionando conectividad IPv6 a usuarios que ya tienen conectividad IPv4

El "tunnel Broker" es el lugar donde el usuario se conecta para registrar y activar su túnel. El Broker gestiona (crea, modifica, activa y desactiva) el túnel en nombre del usuario.

El "tunnel Server" es un router con pila doble , conectado a internet, que siguiendo las ordenes del "broker" crea, modifica o borra los servicios asociados a un determinado túnel /usuario.

El mecanismo para su configuración es tan sencillo como indicar, en un formulario web, datos relativos al Sistema Operativo, la dirección IPv4, un "apodo" para la máquina, y el país don esta conectada. El servidor de túneles crea los registros DNS , el extremo final del túnel, y genera un script para la configuración del cliente*.

* Para mayor información consultar las siguientes páginas web: http://www.freenet6.net http://carmen.cselt.it/ipv6/download.html

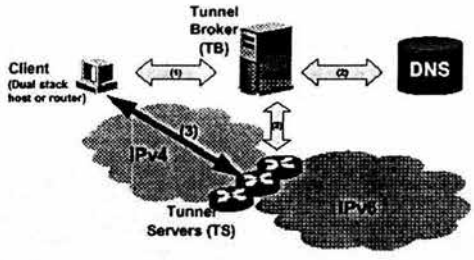


Figura 2.28 Túnel Broker

A continuación se dá un ejemplo, en la figura 2.29, de algunos túneles configurados por la UNAM hacia diferentes clientes

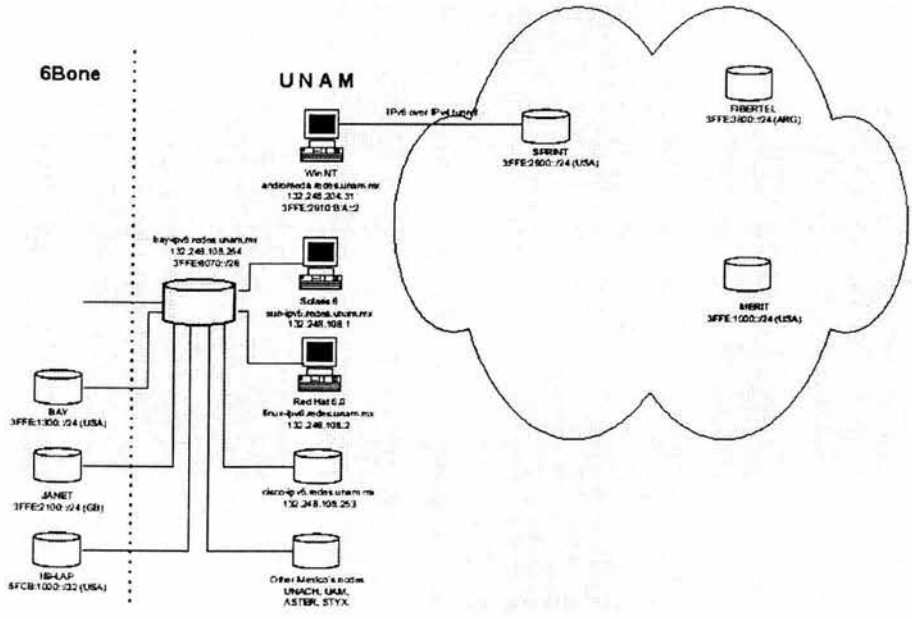


Figura 2.29 Túneles configurados por la UNAM hacia diferentesclientes

2.9 Ruteo

Básicamente se adoptan los mismos protocolos de encaminamiento que los existentes en las redes IPv4, RIP OSPF y BGP. Uno o dos de estos protocolos se utilizará para la implementación de IPv6 en la FES-Cuautitlán.

2.9.1 RIPng

La especificación del protocolo de Información de Rutas (Routing Information Protocol, RIP) para IPv6, recoge los cambios mínimos e indispensable al RFC1058 y RFC1723 para su adecuado funcionamiento.

RIPng es un protocolo pensado para pequeñas redes y por tanto se incluye en el grupo de protocolos de pasarela interior (Interior Gateway Protocol, IGP) y emplea un algoritmo denominado Vector- Distancia. Se basa en el intercambio de información entre routers de forma que puedan calcular las rutas más adecuadas, de forma automática.

RIPng sólo puede ser implementado en routers, donde requerirá como información fundamental, la métrica o número de saltos (entre 1 y 15) que un paquete ha de emplear, para llegar a determinado destino. Cada sitio supone un cambio de red, por lo general atravesando el nuevo router.

Además de la métrica, cada red tendrá un prefijo de dirección destino y la longitud del propio prefijo. Estos parámetros han de ser configurados por el administrador de la red.

El router incorporará, en la tabla de ruteo, una entrada para cada destino accesible (alcanzable) por el sistema. Cada entrada tendrá como mínimo, los siguientes parámetros:

El prefijo IPv6 del destino
La métrica (número de saltos entre este router y el destino)
La dirección IPv6 del siguiente router, así como la ruta para llegar a él
Un indicador relativo al cambio de ruta
Varios contadores asociados con la ruta.

Además se podrán crear rutas internas (saltos entre interfaces del propio router) o rutas estáticas (definidas manualmente)

RIPng es un protocolo basado en UDP. Cada router tiene un proceso que envía y recibe datagramas en el puerto 521 (puerto RIPng)

El inconveniente de RIPng, al igual que en IPv4, siguen siendo, además de su orientación a pequeñas redes (diámetro de 15 saltos como máximo), en que su métrica es fija, es decir, no puede variar en función de circunstancias de tiempo real (retardos, fiabilidad, carga, etc.)[27].

[27] Doylee Jeff, Routing TCP/IP, p. 691-697

2.9.2 OSPFv6

El protocolo de encaminado "Abrir Primero el Camino más Corto" (Open Shortest Path First, OSPF), es también un protocolo de IGP (para redes autónomas), basado en una tecnología de "estado de enlaces" (link-state).

Se trata de un protocolo de ruteo dinámico, que detecta rápidamente cambios de la topología (como un fallo en un router o interfaz) y calcula la siguiente ruta disponible (din bucles), después de un corto período de convergencia con muy poco tráfico de routing.

Cada router mantiene una base de datos que describe la topología del sistema autónomo (de la red) y es lo que denominamos base de datos de "estado de enlaces". Todos los routers del sistema tienen una base de datos idéntica, indicando el estado de cada interfaz y de cada "vecino alcanzable".

Los routers distribuyen sus "estados locales" a través del sistema autónomo de la red por medio de desbordamientos (flooding)

Todos los routers utilizan el mismo algoritmo, en paralelo, y construyen un árbol de las rutas más cortas, como si fueran la raíz del sistema. Este árbol de "rutas más cortas" proporciona la ruta a cada destino del sistema autónomo.

Si hubiera varias rutas de igual coste a un determinado destino, el tráfico es distribuido equilibradamente entre todas. El coste de una ruta se describe por una métrica simple, sin dimensión.

Se pueden crear áreas o agrupaciones de redes, cuya topología no es retransmitida al resto del sistema, evitando tráfico de routing innecesario.

OSPF permite el uso de máscaras diferentes para la misma red (variable length subnetting) lo que permite el encaminamiento a las mejores rutas (las más largas o más especificas)

Todos los intercambios de protocolo OSPF son autenticados y por tanto sólo pueden participar los router verificados (trusted)

OSPFv6 mantiene los mecanismos fundamentales de la versión para IPv4, pero se han tenido que modificar ciertos parámetros de la semántica del protocolo, así como el incremento del tamaño de la dirección. OSPFv6 se ejecuta basado en cada enlace, en lugar de en cada subred.

Además, ha sido necesario eliminar la autenticación del protocolo OSPFv6, dado que IPv6 incorpora estas características (AH y ESP)

A pesar de la mayor longitud de las direcciones, se ha logrado que los paquetes OSPFv6 sean tan compactos como los correspondientes para IPv4, eliminando incluso algunas limitaciones y flexibilizando la manipulación de opciones[28].

2.9.3 BGP4+

El protocolo de pasarelas de Frontera (Border Gateway Protocol, BGP) es un protocolo de encaminado para la interconexión de sistemas autónomos, es decir, para el ruteo entre diferentes dominios.

Frecuentemente se emplea para grandes corporaciones y para la conexión entre proveedores de servicio (ISPs).

Su principal función es, por tanto, el intercambio de información de disponibilidad o alcance entre varios sistemas BGP incluendo información de los sistemas autónomos que contienen, permitiendo así construir las rutas más adecuadas y evitar bucles de tráfico.

BGP4 incorpora mecanismos para soportar ruteo entre dominios sin clases (Classless Inter-Domain Routing, CIRD), es decir, el uso de prefijos, agregación de rutas y todos los mecanismos en los que se basa IPv6.

BGP4 se basa en que un dispositivo sólo informa a los otros dispositivos que se conectan a él, acerca de las rutas que el mismo emplea. Es decir, es una estrategia de "salto a salto". La implicación es la simplicidad de internet, pero la desventaja es que este mecanismo impide políticas complejas, que precisan de técnicas como el ruteo de fuente (source routing)[29].

BGP usa TCP como protocolo de transporte, a través del puerto 179. BGP4+ añade a BGP (RFC1771), extensiones multiprotocolo, tanto para IPv6 como para otros protocolos, como por ejemplo IPX.

Este protocolo será utilizado en la implementación de IPv6 en la parte de ruteo, para que anuncie las redes que se vayan a configurar, así como, recibir todas las redes que están en el ruteador de IPv6 de la UNAM.

2.10 Formato URL

Formato para la representación en URL

Cuando navegamos, continuamente aludimos a URL (de sus siglas en Ingles, Uniform Resource Locator) en muchas ocasiones sin conocer el significado precios de esta abreviatura. La especificación original (RFC2396), nos dice que

[28] Doylee Jeff, Routing TCP/IP, p. 697-704
[29] Ibid, p. 704-78

Localizador de Recurso Uniforme (*Uniform Resource Locator*) es un medio simple y extensible para identificar un recurso a través de su localización en la red. Una vez aclarado esto, y de la misma forma que en ocasiones usamos direcciones en formato IPv4 para escribir un URL, se han descrito unas normas para realizar la representación literal de direcciones IPv6 cuando se usan herramientas de navegación WWW. El motivo por el que ha sido preciso realizar esta definición es bien simple. Con la anterior especificación no estaba permitido emplear el carácter ":" en una dirección, sino como separador de "puerto". Por tanto, si se desea facilitar operaciones tipo "cortar y pegar" de forma rápida para trasladar direcciones entre diferentes aplicaciones era preciso buscar una solución que evitase la edición manual de las direcciones IPv6.

La solución es bien sencilla: el empleo de los corchetes ("[","]") para encerrar la dirección IPv6, dentro de la estructura habitual del URL.

Se ponen algunos ejemplos con las siguientes direcciones:

- FEDC:BA98:7654:3210:FEDC:BA98:7654:3210
- 1080:0:0:0:8:800:200C:4171
- 3ffe:2a00:100:7031::1
- 1080::8:800:200C:417A - ::192.9.5.5 - ::FFFF:129.144.52.38
- 2010:836B:4179::836B:4179

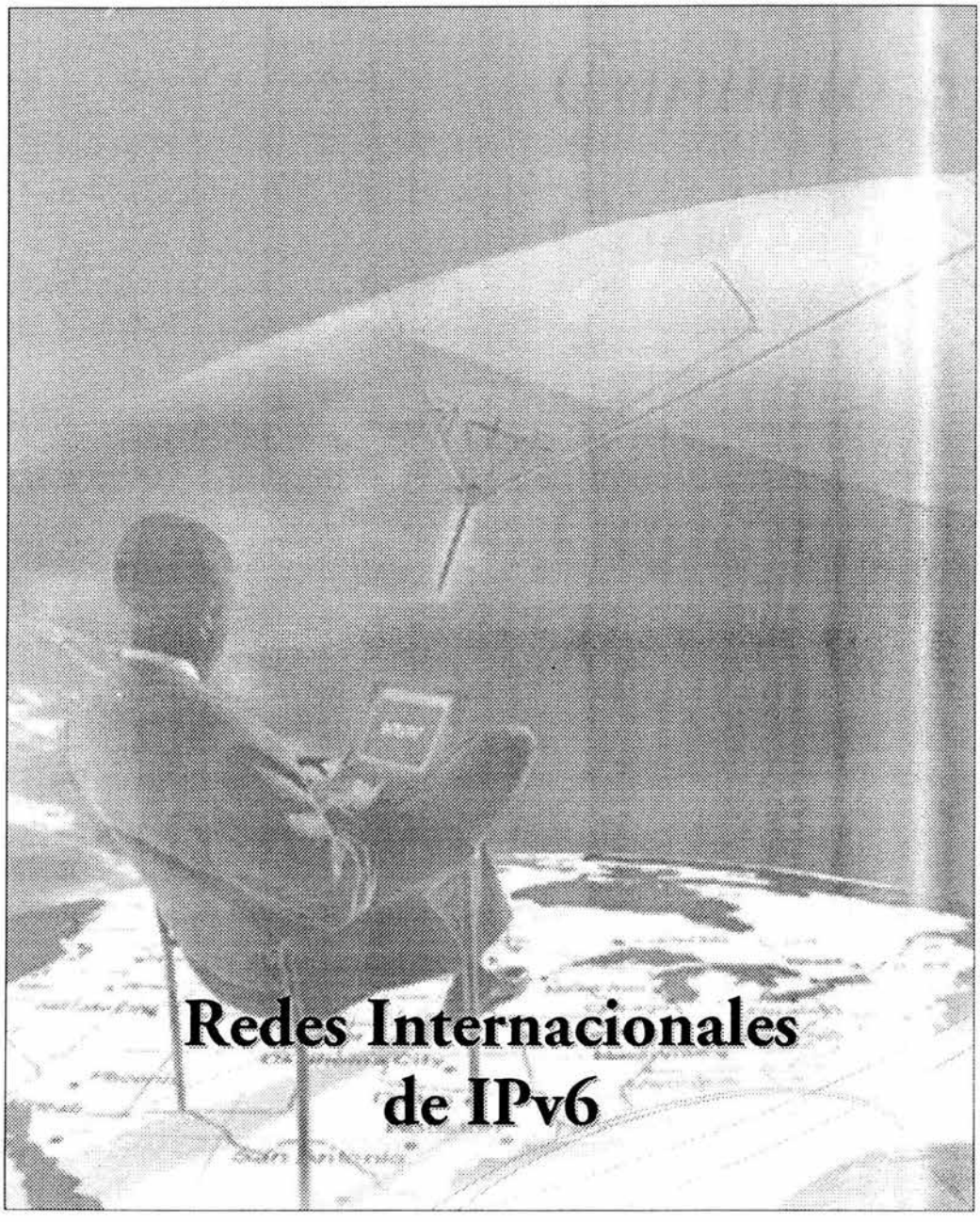
serían representadas como:

- http://[FEDC:BA98:7654:3210:FEDC:BA98:7654:3210]:80/index.html
- http://[1080:0:0:0:8:800:200C:417A]/index.html
- http://[3ffe:2a00:100:7031::1]
- http://[1080::8:800:200C:417A]/foo
- http://[::192.9.5.5]/ipng
- http://[::FFFF:129.144.52.38]:80/index.html
- http://[2010:836B:4179::836B:4179]

Aunque puede parecer un esquema muy complejo, en realidad es muy simple y sobre todo, muy eficiente.

El RFC2450 propone las reglas para la administración de los TLA's y NLA's. Además, en http://www.arin.net/regserv/ipv6/IPv6.txt. podemos encontrar más información al respecto de las normas para registros IPv6, del mismo modo que en http://www.ripe.net/ripencc/about/regional/maps/ipv6policy-draft-090699.html.o en http://www.apnic.net/drafts/ipv6/ipv6-policy-280599.html. En todos los casos, la máxima autoridad competente es IANA (Internet Assigned Numbers Authority).

Capítulo 3



Redes Internacionales de IPv6

CAPITULO 3

Redes Internacionales de IPv6

3.1 6BONE

Derivado del proyecto IPv6 de la IETF nace 6Bone, una red mundial experimental usada para probar los conceptos e implementaciones de IPv6.

6Bone es una red compuesta por "islas" que soportan IPv6, unidas por enlaces punto a punto llamados "túneles", y opera según el esquema de direcciones experimental establecido en el RFC 2471: *IPv6 Testing Address Allocation*

Los pasos que se realizaron para establecer un nodo de 6Bone en la UNAM consistieron en:

1. Solicitar, configurar y probar un túnel de IPv6 sobre IPv4 a un nodo ya conectado a 6Bone.

2. Dar de alta la información del nodo activo de la UNAM en la base de datos del 6Bone.

A continuación se presenta un registro de entrada al 6BONE

Registro de entrada al 6BONE

Aquí se le indica lo que se debe hacer para crear una entrada en el registro de 6BONE para estar conectado a la red mundial de IPv6; tener registrado el bloque de direcciones asignadas y dar de alta referencias propias para la resolución de problemas. Existe una referencia general en 6bone Registry Documentation*, que indica como crear varias entradas que recomendamos revisar ya que es mas general. Aquí se especifica un método por e-mail para crear/actualizar entradas de registro en 6BONE en forma rápida, sin ser el método preferido para usar.

El método preferido de creación/actualización de entradas en el registro 6BONE es el de Viagenie, 6BONE REGISTRY DATABASE WEB INTERFACE*. Este método proporciona una protección para la autorización, al requerir que todas las entradas sean protegidas por entradas **mntner** con un esquema de autenticación de contraseña encriptada.

Los pasos mínimos necesarios para dar de alta en forma oficial el bloque delegado, son:

---

* consultar la página http://6bone.net/RIPE-registry.html
* consultar http://www.viagiene.qc.ca/en/ipv6/registry/index.html

1.- Crear el un objeto PERSONAL. Dado que todavía no se tiene un objeto MNTNER, se deja en blanco el campo *mnt-by*. Si los datos ingresados fueron correctos recibirá un mensaje de OK. de creación del objeto. Si no, checar la sintaxis de los campos indicados con ERROR.

2.- Crear un objeto de MNTNER. El nombre que se elija será el mismo que se usará para el campo MNTNER. Recibirá un mensaje de advertencia (el cual se parece a un mensaje de error) diciendo que su petición se manejará manualmente.

3.- Esperar un correo electrónico informando acerca de la creación del objeto MNTNER (esto puede tomar varias horas).

4.- Cuando se tenga un objeto MNTNER editar su objeto personal (creado en el primer paso) y llenar el atributo mnt-by usando el objeto MNTNER creado en el paso #2.

5.- Se envía un correo dando aviso de los datos dados de alta y esperar un correo de confirmación de la creación del objeto *inet6num*, que es el Bloque de direcciones delegado para la UNAM, y de la creación de un túnel (si este fue solicitado).

6.- Finalmente lo único que resta, es consultar la pagina web de Viagenie 6BONE REGISTRY DATABASE WEB* y continuar con el registro de su bloque de direcciones. Dando de alta el objeto ipv6-site donde se indica entre otros campos, el túnel con nosotros. Ver Ejemplo.

El siguiente ejemplo, corresponde a un ejemplo de registro de la Universidad Nacional Autónoma de México en la base de datos de 6bone, además de los registros de algunos túneles con la misma*.

% RIPEdb(3.0.0b2) with ISI RPSL extensions

```
inet6num:    3FFE:8070::/28
netname:     UNAM
descr:       pTLA delegation for the 6bone
country:     MX
admin-c:     COM1-6BONE
tech-c:      COM1-6BONE
notify:      cesar@redes.unam.mx
mnt-by:      MNT-UNAM
changed:     fink@es.net 19990927
source:      6BONE
```

* ver página web http://www.viagiene.qc.ca/en/ipv6/registry/index.html

* Datos proporcionados por el grupo de trabajo de IPv6 de la UNAM

ipv6-site: UNAM
origin: AS278
descr: Universidad Nacional Autonoma de Mexico 6Bone pTLA Site, Mexico City
country: MX
prefix: 3FFE:8070::/28
application: ping bay-ipv6.redes.unam.mx
tunnel: IPv6 in IPv4 bay-ipv6.redes.unam.mx -> 6bone.merit.edu MERIT STATIC
tunnel: IPv6 in IPv4 bay-ipv6.redes.unam.mx -> ulcc.ipv6.ja.net JANET STATIC
tunnel: IPv6 in IPv4 bay-ipv6.redes.unam.mx -> iosv6-7k.ep.net ISI-LAP RIPng
tunnel: IPv6 in IPv4 bay-ipv6.redes.unam.mx -> ipv6.cic.ipn.mx CIC-IPN STATIC
contact: COM1-6BONE
remarks: Platforms: Bay Networks BLN and Cisco 2600
remarks: Will add new tunnels and delegate address space upon request
remarks: This ipv6-site is operational since June 17, 1999
url: http://www.ipv6.unam.mx
notify: cesar@redes.unam.mx
mnt-by: MNT-UNAM
changed: cesar@redes.unam.mx 19990617
changed: cesar@redes.unam.mx 20001021
source: 6BONE

person: Cesar Olvera
address: Universidad Nacional Autonoma de Mexico
address: DGSCA-UNAM, C. U., Mexico City
phone: +52 5622 8526
e-mail: cesar@redes.unam.mx
nic-hdl: COM1-6BONE
notify: cesar@redes.unam.mx
mnt-by: MNT-UNAM
changed: cesar@redes.unam.mx 19990617
changed: cesar@redes.unam.mx 19990913
source: 6BONE

3.2 6REN

El 6REN (Red IPv6 para Investigación y Educación) es una iniciativa voluntaria de la coordinación de las redes de la investigación y de la educación para promover servicios de red de producción IPv6 que faciliten alta calidad, alto rendimiento, y operacionalmente las redes robustas IPv6.

En octubre de 1998 la "Energy Science Network" (Esnet) estableció el proyecto de 6REN, el cual es un proyecto de redes de investigación y educación para proveer

servicios de tránsito de IPv6. El primer paso de 6REN consistió en establecer interconexiones de IPv6 nativo sobre ATM entre ESnet, Internet2/vBNS, Canarie, Cairn y WIDE.

La participación está libre y abierta a todas las redes de la investigación y de la educación que proporcionen el servicio IPv6*. Cualquier red que proporciona, o que planea proporcionar, los servicios de producción IPv6 puede participar. Cualquier red que proporcione servicios de la calidad IPv6 de la producción se autoriza para demostrar la insignia **6ren** en sus sitios web.

3.3 EURO6IX

Euro6IX (European IPv6 Internet Exchanges Backbone) su finalidad es el soporte de la rápida introducción de IPv6 en Europa. Para la consecución de esta meta, el proyecto ha definido un plan de trabajo con todos los pasos necesarios incluyendo: el diseño de la red Pan-europea (con IPv6 nativo) y su implantación; la investigación de los servicios avanzados de dicha red; el desarrollo de aplicaciones que serán validadas mediante la vinculación de grupos de usuarios y pruebas internacionales; y actividades de diseminación activa, incluyendo eventos y conferencias, contribuciones a estándares (IETF, RIPE y otros), publicación de informes, y promoción de todos los resultados del proyecto a través de la web del mismo*.

El **principal** objetivo del proyecto Euro6IX es la investigación de la arquitectura apropiada, el diseño y la implantación de la primera red Pan-europea, no comercial, de Intercambiadores de tráfico Internet IPv6. Conectará diversos puntos intercambiadores regionales de tráfico IPv6, a través de Europa, y alcanzará el mismo nivel de robustez y calidad de servicio que actualmente ofrecen redes similares IPv4.

La red será la base para la realización de actividades de investigación relacionadas con diversos aspectos de IPv6 como: investigación de la madurez de los servicios avanzados de red, así como la posibilidad de su introducción en la red Euro6IX, como son QoS/CoS, Movilidad IP, Anycast y multicast, seguridad, multihoming, renumeración, y lenguajes de políticas.

La red será ejercitada por los Grupos de Usuarios para validar y verificar el uso, características, y potencial, de la Internet de siguiente generación, en eventos de alta visibilidad y pruebas (tanto internas como públicas).

---

* para mayor información consultar las páginas http://www.6ren.org y http://www.6net.org/overview.html

* Para más detalles consultar http://www.euro6ix.org

3.4 IPv6 en la UNAM

3.4.1 Historia de IPv6 en la UNAM

A continuación se presenta un bosquejo histórico del desarrollo de IPv6 en la UNAM:

- En diciembre de 1998 iniciaron formalmente las pruebas sobre el uso de IPv6.
- En enero del siguiente año (1999) se hizo la petición de un túnel para conexión a la red experimental "6Bone", se realizaron pruebas con Windows NT instalando el stack MSR IPv6 v.1.1 de Microsoft para tener el primer túnel con Sprint. También inició el estudio y prueba de las características del stack IPv6 v. 5.3 para Solaris 2.5 y de aplicaciones IPv6 para UNIX
- Para marzo el mismo año, se dio inicio a una serie de pruebas de conexión con ruteadores BAY en colaboración con Nortel Netwoks.
- En mayo de 1999 se realizaron pruebas con túneles de IPv6 sobre IPv4 con Sprint, EUA y Fibertel, Argentina, usando el stack MSR IPv6 v.1.2.
  Se logró el registro oficial del nodo 6Bone de la UNAM-Nortel Networks, siendo el 1er en su tipo en México.
  También se abrieron páginas Web en IPv6 y fue asignado un túnel automáticamente por Freenet6.net (Canarie), Canadá.
- Posteriormente en junio, se elaboró la primera versión de ésta página Web para IPv4 con información de las pruebas IPv6.
- En julio, se probó la nueva versión del stack MSR IPv6 de Microsoft en Win NT con mecanismos de seguridad integrados, se continuó con el estudio del DNS para IPv6 y de IPv6 sobre ATM. Para finales del mes, se asignó otro túnel automático por CSELT, Italia.
- Para el siguiente mes, agosto, se dio inicio a la construcción de la "RedUNAM IPv6", con la adquisición de un ruteador BLN de Nortel Networks, para correr aplicaciones IPv6 y proveer servicios IPv6 en México y en Latinoamérica.
- En septiembre se hizo la petición para que la UNAM fuera un nodo de Backbone de 6Bone, se estudiaron y probaron las características de un stack y de aplicaciones IPv6 para Linux , y se instaló y configuró un servidor de DNS en IPv6, usando el stack IPv6 v. 5.3 para Solaris 2.5.
  También se establecieron túneles proporcionados por Fibertel Argentina, Merit EUA, la Universidad de Londres en Gran Bretaña, y hacia instituciones en México como la UAM, ASTER México, el IPN y el ITAM.
- Al mes siguiente, se llevaron a cabo presentaciones de la RedUNAM IPv6 en eventos como "Gobierno 2000" y "Computo 99" en la ciudad de México.
  Se impartió el primer curso para la configuración de túneles en Win NT.
- En noviembre de este año se establecieron más túneles con instituciones y empresas alrededor del mundo. Se planeó la integración del IPv6 con la red de Internet 2 de México, y se presento el 1er Seminario Nacional de IPv6.
- En el 2002 se le asigna a la UNAM un bloque de producción prefijo /32 (2001:0048::/32)

En la siguiente figura se muestran las primeras conexiones que tenía la UNAM

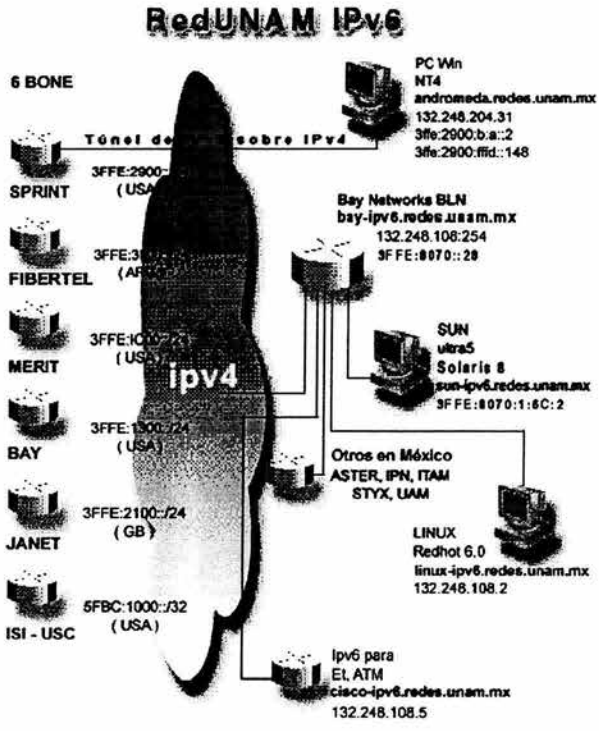


Figura 3.1 Primeras conexiones IPv6 de la UNAM

3.4.2 Red IPv6-UNAM

La UNAM inició investigaciones en la materia desde el mes de diciembre de 1998, fecha en la que se constituye el proyecto IPv6 en nuestra Máxima Casa de Estudios, y durante el segundo semestre del año 1999 es notable el liderazgo de la UNAM en el ámbito nacional.

Dentro de las primeras pruebas realizadas, destaca la de conexión a 6Bone (www.6bone.net), la cual es una red mundial experimental, como se mencionó anteriormente, utilizada para probar los conceptos y la puesta en práctica de IPv6. Actualmente participan varios países, entre ellos México, donde la UNAM fue el primer nodo en México, registrado en junio de 1999. Posteriormente, la UNAM es aceptada como nodo de Backbone de 6Bone en septiembre de 1999, obteniendo un rango de direcciones tipo **pTLA (pseudo Top-level Aggretation): 3ffe:8070::/28**, que ha sido utilizado para pruebas de investigación, instalación y evaluación de IPv6. Posteriormente en octubre del 2000 la UNAM recibió un bloque del tipo **sTLA (sub Top-Level Aggregation): 2001:0448::/35**,y en el 2002

se le delegó a la UNAM un prefijo /32 quedando así: **2001:0448::/32** con esto las comunidades de Internet, tanto de México como de Latinoamérica, tendrán acceso a servicios basados en IPv6 ya que este tipo de direcciones las utilizan compañías como MCI, Verio y Sprint en EU, y NTT en Japón, para ofrecer a sus usuarios algunos servicios de producción con IPv6. De esta forma, la asignación de este bloque de direcciones a la UNAM, es un importante paso en la instalación y uso de IPv6 en México. Además ahora la UNAM puede delegar direcciones y agregar túneles a instituciones en el mundo interesadas en realizar pruebas con IPv6.

Los campos donde se aplica y se aplicará IPv6 son: administración de redes, seguridad en redes, supercómputo, educación a distancia, sistemas de información geográfica, telemedicina, laboratorios virtuales, bibliotecas digitales, realidad virtual, entre otros.

3.4.2.1 Proyecto IPv6 de la UNAM

La UNAM tomando el liderazgo en la instalación de IPv6 en México, da inicio al Proyecto IPv6 de la UNAM en diciembre de 1998, los objetivos generales del proyecto son:

- Investigar, probar e implementar el protocolo IPv6 en la Red Integral de Telecomunicaciones de la UNAM.
- Participar en el desarrollo de proyectos de IPv6 nacionales e internacionales.
- Participar en el fortalecimiento y difusión de IPv6 y sus aplicaciones.
- Proveer servicios de IPv6 en México y Latinoamérica.
- Evaluar e implementar el protocolo IPv6 en la Red Internet2 de México.

Dentro del Proyecto IPv6 de la UNAM se estableció un amplio programa de pruebas y trabajos con temas como: implementaciones, stacks IPv4/IPv6, túneles, software de conexión, aplicaciones multimedia, Web, DNS, autoconfiguración, calidad de servicio, IPv6 sobre ATM, redes internacionales de IPv6 (6Bone, 6REN), IPv6 en Internet2, etc*.

Para contar con una **red de pruebas** en una primera etapa, y posteriormente con una **red de producción**, se instaló la Red IPv6 de la UNAM, **la primera red IPv6 instalada en México** y que inició operaciones en agosto de 1999. Esta red cuenta con varios túneles hacia otros nodos de Backbone de 6Bone: SPRINT, FIBERTEL, MERIT, BAY NETWORKS, JANET e ISI-LAP, y hacia los hosts que tiene la UNAM corriendo con sistemas operativos como Win NT Solaris y Linux. Actualmente se esta trabajando con instituciones mexicanas para realizar su conexión IPv6 hacia la UNAM, entre estas instituciones destacan: Instituto Politécnico Nacional, Universidad Autónoma Metropolitana, Instituto Tecnológico de Estudios Superiores de Monterrey, Universidad Autónoma de Chiapas, Universidad Autónoma de

* La pagina http://www.ipv6.unam.mx contiene mas información sobre los trabajos realizados.

Guerrero, Instituto Tecnológico de Oaxaca, Instituto Tecnológico de Mérida, Petróleos Mexicanos, STYX, ASTER, etc*.

En la figura 3.2 se muestra la red de pruebas IPv6 de la UNAM

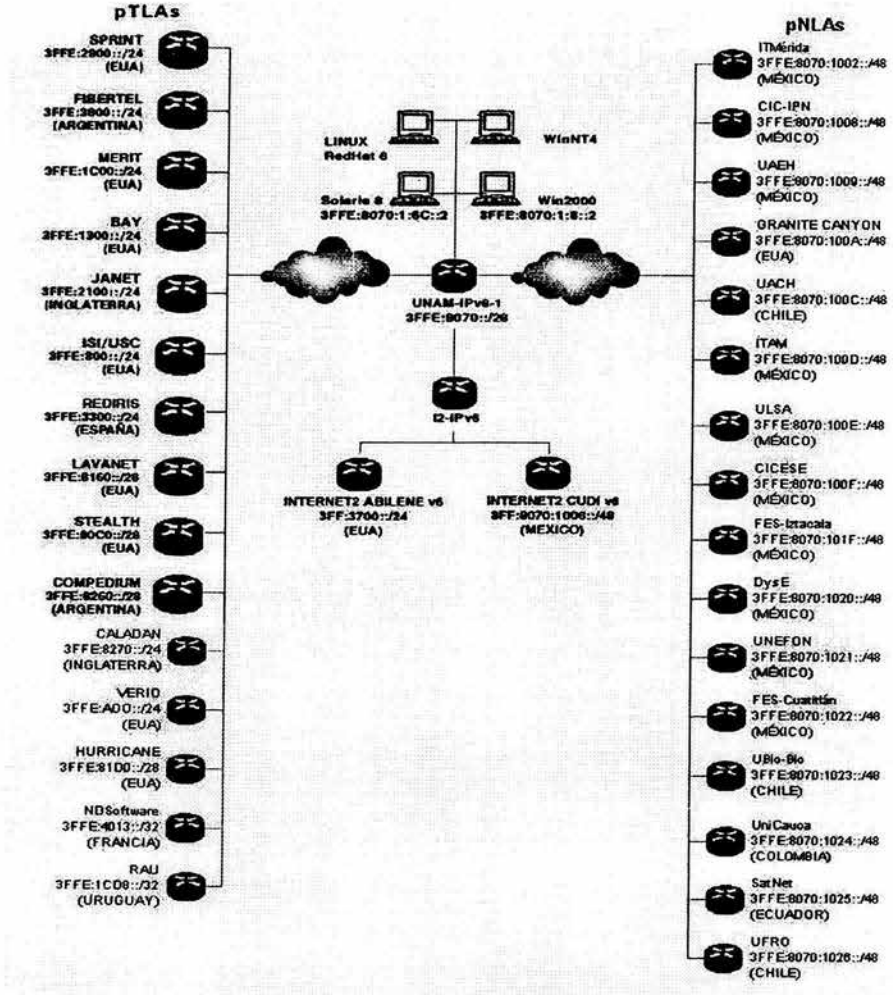


Figura 3.2 Red de pruebas IPv6 de la UNAM

En la figura 3.3 se ilustra la red de producción IPv6.

* Para información adicional relacionada con el desarrollo del proyecto IPv6 en México visite la página Web http://www.ipv6.unam.mx

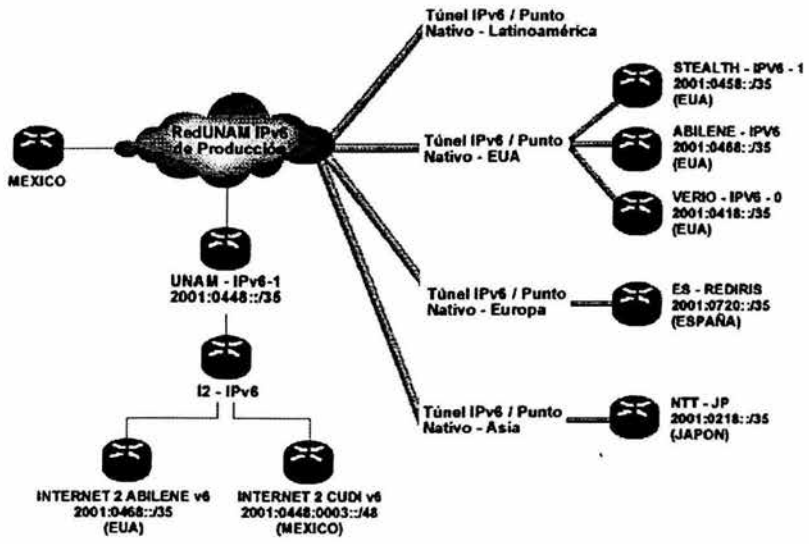


Figura 3.3 Red de producción de la UNAM

Entre los trabajos de difusión del Proyecto IPv6 de la UNAM, esta la presentación de los principales resultados en Congresos, Seminarios y reuniones nacionales en internacionales, pudiendo mencionar los siguientes:

- Pruebas IPv6.
  Presentado en: DGSCA-UNAM, 5 agosto 1999.

- RedUNAM IPv6.
  Presentado en: Cómputo.99@mx, 8 octubre 1999.

- Review of UNAM`s IPv6 Project.
  US IPv6 Summit, Telluride Colorado, Marzo 2000.

- IPv6 Network of the UNAM.
  International Conference on Telecommunications ICT 2000, Acapulco México, Mayo 2000.

- Tutorial IPv6
  César Olvera Morales, diciembre 2000.

Como bien se ha mencionado anteriormente la instalación de IPv6 en México se dio principalmente en la UNAM seguido por otras universidades, que conjuntamente han hecho de este proyecto una Red Nacional de IPv6.

En la figura 3.4 se ilustra un mapa de la red IPv6 en México.

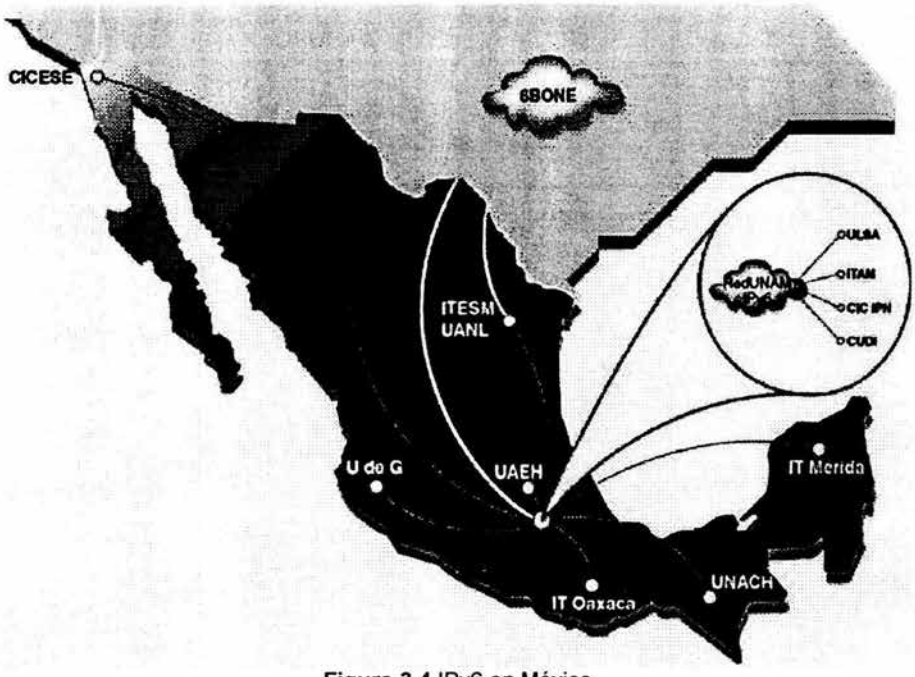


Figura 3.4 IPv6 en México

3.4.3 Red CUDI

Corporación Universitaria para el Desarrollo de Internet (CUDI)

A mediados del año de 1997 surgió la inquietud de algunas instituciones de educación superior en México de crear una infraestructura de comunicaciones capaz de ofrecer un servicio de conectividad para el desarrollo de nuevas tecnologías de Internet, a la par con las iniciativas Internet 2 que se iban dando en otros paises, UCAID en los E.U. y CANARIE en Canadá, en México se consolido CUDI Corporación Universitaria para el Desarrollo de Internet, con lo que fue posible crear un grupo de trabajo cuyo objetivo principal fue organizar la instalación de una mesa de pruebas por medio de la cual se pudiesen implementar aplicaciones avanzadas de Internet, así como brindar conectividad con otras iniciativas similares en otros países.

Ya entonces era evidente que las aplicaciones serían las que marcarían la pauta para el desarrollo de una red tipo Internet 2. Servicios como videoconferencia interactiva, vídeo en demanda, herramientas de colaboración, laboratorios

remotos, control de aplicaciones en tiempo real, etc. requieren que la red responda de manera particular a cada una de ellas, mínimo tiempo de respuesta (low latency) y alta disponibilidad. Esto se lograba implementar en "ambientes controlados" algunos de estos "ambientes controlados" se extendían ya como redes metropolitanas o redes privadas en algunas instituciones de educación superior. La red CUDI describe como un elemento de conectividad que permite a las redes de comunicación de alta capacidad de las instituciones de educación superior Mexicanas la comunicación entre ellas así como con las otras iniciativas internacionales de conectividad del tipo Internet 2.

Como resultado de diversas reuniones entre representantes de las Universidades miembros de CUDI se presenta. **El proyecto nacional de conectividad Internet 2** que tiene por objetivo el **Diseño e Implementación de la Red Internet 2 en México**.

Diseño de la Red.

Son identificadas las necesidades particulares de 8 distintos tipos de aplicaciones que se utilizarán en la red:

1.- transferencia de archivos y cómputo distribuido.
2.- Visualización para supercómputo.
3.- Sesiones remotas para supercomputadoras (supertelnet).
4.- Telepresencia Audio.
5.- Telepresencia Vídeo.
6.- Telepresencia AV (y tacto).
7.- Hipermedia.
8.- Control de aplicaciones en tiempo real. Cada uno de estos tipos de aplicaciones demandará recursos a la red en una forma particular y diferente una de las otras.

Se define posteriormente utilizar una arquitectura tecnológica para la red basada en el protocolo IP sobre una red de conmutación de celdas ATM que permitirá incorporar la mayor cantidad de aplicaciones a esta red. Aplicaciones de Internet sin controles avanzados, aplicaciones de Internet con controles avanzados, aplicaciones de red basadas en la conmutación de circuitos (Videoconferencias H.320), multicast.

Se propuso en conjunto con TELMEX un diseño preliminar para el backbone de CUDI, que permitirá conectar las regiones del País con enlaces de STM-1 de 155 Mbps. Así mismo se divide la operación de la red en dos niveles 1. Nivel de backbone y nivel de acceso.

En base a la propuesta de TELMEX para proveer una infraestructura de comunicaciones y equipo como participación en calidad de Asociado Institucional en la Corporación Universitaria para el Desarrollo de Internet, el nivel de backbone se construyó en las ciudades de Guadalajara, México, Monterrey y Tijuana con

dos conexiones Internacionales una en la ciudad de San Diego California y la otra en la ciudad de El Paso Texas unidas a la red de CUDI por medio de enlaces de una capacidad no inferior a los 155 Mbps conectando en los nodos de backbone de Tijuana y Monterrey respectivamente.

A partir de estos nodos de backbone se incorporarán Nodos de Acceso que atenderán las solicitudes de conexión de los Afiliados a CUDI en las distintas regiones del país. Los Nodos de Acceso son integrados a la red de CUDI con un enlace no inferior a los 34 Mbps y deberán contar con una infraestructura de comunicaciones, de recursos humanos y general que permita ofrecer el servicio de conexión a los afiliados asegurando un mínimo de fallos y un "uptime" superior al 99.5%

Los nodos de acceso iniciales y para los cuales se cuenta con el financiamiento para su operación son: Universidad de Guadalajara, Instituto Tecnológico de Estudios Superiores de Monterrey campus Monterrey, Universidad Autónoma de Nuevo León , Universidad Autónoma de Tamaulipas, Instituto Politécnico Nacional, Universidad Nacional Autónoma de México, Universidad Autónoma Metropolitana, Universidad Autónoma de Puebla y el Centro de Investigación Científica y de Estudios Superiores de Ensenada.

La designación de Nodos de Acceso adicionales deberán responder a las necesidades particulares de una región del país y deberán ser promovidas por el comité para el desarrollo de la red para su aprobación por el Consejo Directivo de CUDI exclusivamente. Esto para asegurar un crecimiento ordenado de la red y niveles de calidad de servicio óptimos para los afiliados de CUDI.

De acuerdo al diseño de la red los nodos de acceso deberán ofrecer el servicio de conectividad en los niveles de capa 2 y 3 del modelo OSI. A nivel de capa dos por medio de puertos de acceso en ATM y en el nivel 3 con puertos de acceso y protocolo IP versión 4.

Los Afiliados deberán conectarse a los nodos de acceso a través de enlaces por lo menos de 2 Mbps en inicialmente se brindara el servicio de conectividad en IP solamente. En la siguiente figura se muestra la red de CUDI.

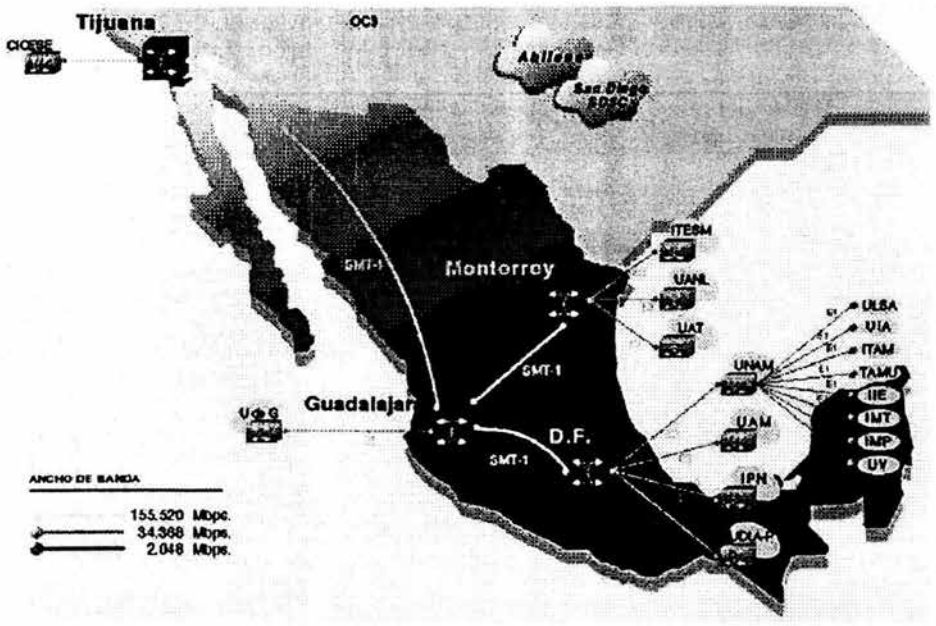


Figura 3.5 Red CUDI

La Administración de la Red (Gestión de Red)

TELMEX será responsable de la gestión de la *red física* en las 5 áreas definidas por la OSI. Se entiende por red física la infraestructura de hardware necesaria para interconectar los "Gigapops" sus características físicas, eléctricas y de procedimiento. También la configuración general de equipos de backbone, esto es ubicación física, instalación de interfaces y/o puertos, etc.

CUDI será responsable de la gestión de la *red de servicios* en las 5 áreas definidas por la OSI*. Se entiende por red de servicios todo aquello que tenga que ver con el establecimiento de controles avanzados sobre la infraestructura de red física, Priorización de tráfico establecimientos de circuitos virtuales permanentes y conmutados, actualizaciones de sistema operativo en equipos de backbone, "configuration tunning", entre otros.

Equipamiento.- Los Nodos de Backbone contarán con equipos CISCO BPX 8650 equipados con interfaces STM-1 y E3 para poder asegurar la operación como Switches ATM + Servicios de ruteo de IP.

* Red física = Capa 1 del modelo OSI
Red de Servicios = Capas 2 en delante del modelo OSI + Sistema operativo

Los nodos de Acceso deberán contar con equipos independientes de la red interna de su respectiva institución para brindar el acceso. Un switch ATM para recibir el enlace E3 (34Mbps) y un ruteador para brindar el acceso a la Red con enlaces de por lo menos 2 Mbps. En la figura 3.6 se muestra el backbone existente en CUDI.

El equipamiento de los nodos de acceso será responsabilidad de los mismos apoyados por los Asociados académicos e institucionales de CUDI.

El 8 de abril de 1999 se constituye oficialmente la Corporación Universitaria para el Desarrollo de Internet (CUDI)*, en un esfuerzo conjunto del gobierno mexicano, la comunidad universitaria y la sociedad mexicana en general, con el fin de desarrollar una red de alta velocidad y unirse a la red internacional Internet2.

3.5 Internet2 e IPv6 en la UNAM

Internet2 es una red experimental cuya meta es buscar la colaboración entre universidades y empresas para desarrollar tecnología en redes y aplicaciones que complementen la misión de investigación y educación de las universidades. Desde que la enseñanza, el aprendizaje y la investigación han empezado a requerir más elementos de multimedia en tiempo real y con un ancho de banda mayor, el objetivo de Internet2 ha sido crear la infraestructura necesaria en redes para correr aplicaciones tales como: telemedicina, bibliotecas digitales, laboratorios virtuales, educación a distancia, entre otros. Aunque Internet2 no esta considerado en un corto plazo como un reemplazo de Internet, los participantes esperan poder compartir sus desarrollos con los usuarios de Internet en un futuro cercano.

Como parte de los trabajos de CUDI en la red Internet2 en México y la Universidad Nacional Autónoma de México (UNAM), quien coordina el grupo de trabajo de IPv6 en CUDI, a finalizado la fase de instalación y conexiones usando el nuevo protocolo IPv6. Con esto se avanza en las pruebas e implantación de nuevas tecnología que permitirán llevar a cabo las aplicaciones planeadas para Internet2.

Como vimos en el apartado anterior, la Corporación Universitaria para el Desarrollo de Internet (CUDI), constituida por las principales universidades de México, es quien coordina el proyecto de Internet2 en nuestro país. La misión de CUDI es la de promover y coordinar el desarrollo de redes de telecomunicaciones y cómputo, enfocadas al desarrollo científico y educativo en México. Para lograr esto se trabaja sobre los siguientes objetivos específicos: Promover la creación de una red de telecomunicaciones con capacidades avanzadas. Fomentar y coordinar proyectos de investigación para el desarrollo de aplicaciones de tecnología avanzada de redes de telecomunicaciones y cómputo enfocadas al desarrollo científico y educativo de la sociedad mexicana.

* Para mayor información sobre CUDI consultar la página web: http://www.cudi.edu.mx

El proyecto de instalación de IPv6 en la red de Internet2 de CUDI, que inició en abril del 2001, contempló la implantación de IPv6 en el backbone de la red de Internet2, con cuatro nodos en Ciudad de México, Guadalajara, Monterrey y Tijuana; además de implantarlo en los equipos de acceso de las universidades conectadas a la red de CUDI. Este trabajo de IPv6 lo coordina la UNAM. Cabe destacar que este proyecto de instalación de IPv6 en Internet2 es el proyecto más importante en su tipo en México*.

Actualmente Internet2 cuenta con las conexiones que se muestran en la figura 3.6

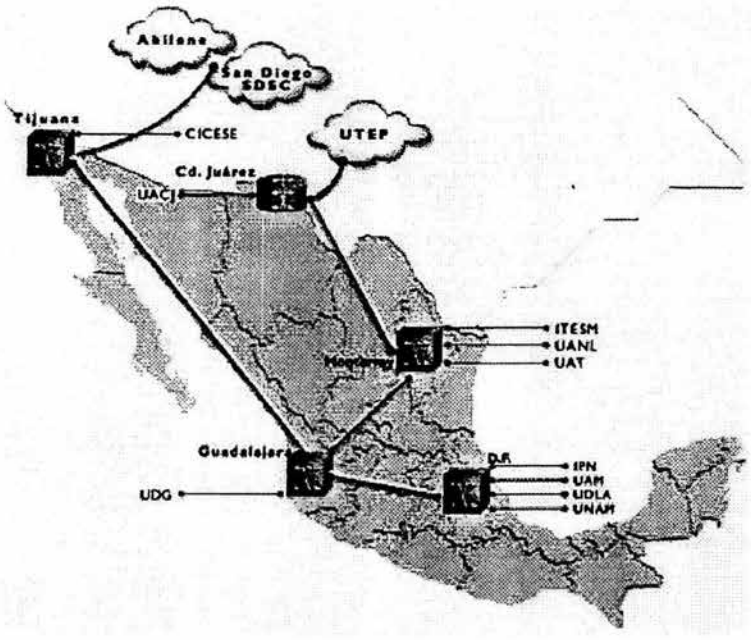


Figura 3.6 Red Internet2-México

En un principio el backbone de Internet2 en México estaba constituido por las siete universidades fundadoras del proyecto: Universidad Nacional Autónoma de México, Instituto Politécnico Nacional, Universidad Autónoma Metropolitana, Instituto Tecnológico de Estudios Superiores de Monterrey, Universidad Autónoma de Nuevo León, Universidad de la Américas, Universidad de Guadalajara, y por el Centro de Investigación Científica y Educación Superior de Ensenada y la Universidad Autónoma de Tamaulipas.

* Para información adicional relacionada con el desarrollo del proyecto IPv6 en México, tanto en Internet como en Internet2 visite las páginas Web: http://www.ipv6.unam.mx, http://www.internet2.unam.mx y http://www.noc-internet2.unam.mx

En la figura 3.7 se muestra un esquema de direccionamiento del backbone IPv6 utilizando Internet2

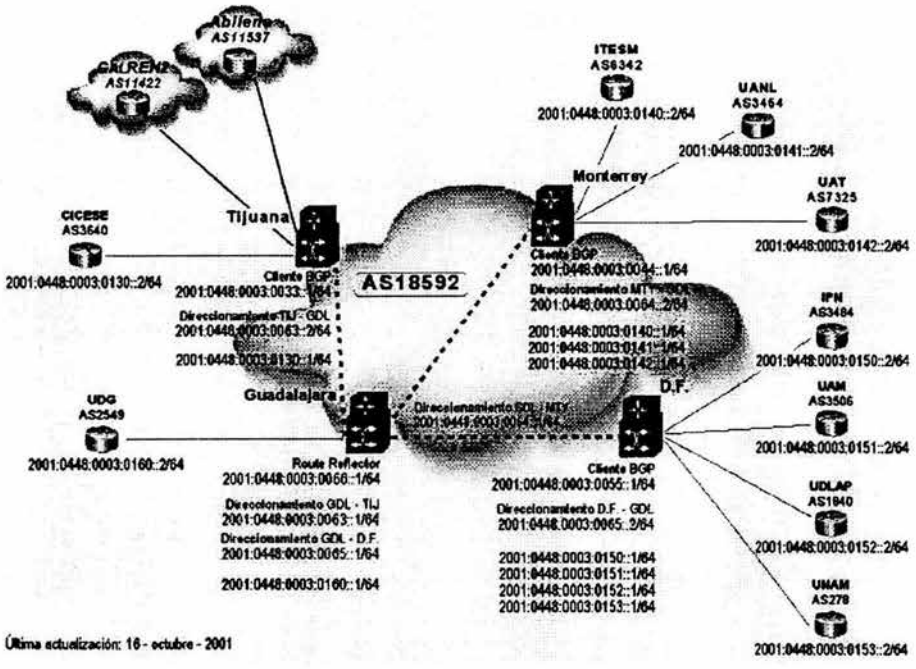


Figura 3.7 Backbone IPv6

Centro de Operación de la Red Internet2 de México (NOC-I2)

El proyecto de Internet2 tiene como objetivo principal impulsar el desarrollo de aplicaciones académicas que requieren una infraestructura de red de alto desempeño que faciliten las tareas de investigación y educación de las universidades y centros de investigación. Simultáneamente el proyecto aumentará las potencialidades de las redes multimedia de banda ancha que seguramente en un plazo no muy lejano estarán disponibles al resto de la comunidad de Internet.

Así, Internet2 proporcionará un marco para el desarrollo de aplicaciones de vanguardia tales como realidad virtual, colaboración remota, bibliotecas digitales, laboratorios virtuales, por mencionar algunas. Las características de estas aplicaciones demandarán el incremento en el control y calidad de los medios y equipos de comunicación, desarrollando y aplicando tecnologías de vanguardia como ingeniería de tráfico, calidad de servicio, multicast, etc., así como también robustecer los esquemas de monitoreo y administración de estas redes.

La red Nacional Internet2 se encuentra en el Centro de Operación en la UNAM. Este Centro de Operación de Red (Network Operation Center, NOC) es el centro encargado de mantener la operación de la red en óptimo desempeño y funcionalidad. Sus actividades aseguran una alta disponibilidad del servicio, un rápido reconocimiento de fallas y detección de niveles de degradación del servicio. Entre sus tareas se abarcan actividades de control proactivas y correctivas, así como la coordinación de pruebas tecnológicas con otros grupos de trabajo inherentes al desarrollo de la red.

El NOC divide sus funciones en: monitoreo, operación eficiente e integridad de todos los dispositivos que conforman la red, atención y seguimiento de fallas, emisión de estadísticas, soporte técnico a los administradores de red de cada institución que conforma la red, mantenimiento de las bases de datos e inventarios de equipos. Además de sus responsabilidades directas, el NOC deberá también contribuir con los grupos encargados de análisis, planeación, seguridad, procedimientos y pruebas de tecnologías emergentes.

La responsabilidad de la UNAM como NOC de la Internet2 mexicana es implantar un esquema de operación y monitoreo adecuado y eficiente que englobe a todos los participantes en el proyecto: Nortel, Fore, Cabletron, entre otros, bajo sistemas de gestión basados en RMON y SNMP, en plataformas robustas de cómputo y telecomunicaciones, dando de esta manera respuesta a los niveles de calidad e integración que los proyectos de la comunidad de investigadores y académicos nacionales demandará*.

3.6 Aplicaciones sobre IPv6

El número de aplicaciones a IPv6 es escasísimo el día de hoy. No obstante, podemos probar con las que tenemos, por ejemplo, a) Navegadores web: mozilla, galeon, b) Servidores web: apache, b) Juegos: Quake II, tetris, c) Otros: SSH, xchat, etc

Portar aplicaciones a IPv6 no es excesivamente complicado en el caso de aplicaciones *bien diseñadas*. Enseguida mencionaré *algunas aplicaciones hoy en día:

Chat software

- IRC: Bitchx client – Ahora soporta IPv6
- RAT and SDR – versión de windows el UCL puertos de conferencia

---

* Referencias http://www.internet2.edu.mx, http://www.dtd.unam.mx, http://www.noc.unam.mx

* para mayor información y actualización de las aplicaciones consultar la pagina web http://www.ipv6forum.com

DNS

- BIND 9.1.2 – los usos de registro A6 y ofrecen el transporte IPv6
- Totd – un DNS poxy para soportar la transición IPv4/IPv6 (traducción)
- IPv6 transport for BIND8 – un parche para BIND 8.2.3 por Stig Venas

Firewalls

- CheckPoint - planes para FireWall-1 en IPv6
- Firewalling in OpenBSD - una guía del instituto SANS
- ipfilter – soporta filtros IPv6
- IPFW – incluido dentro del Free BSD 4.0
- Netfilter – parches IPv6 para Linux

FTP

- LFTP – soporta IPv6 nativo
- NcFTP (Windows) – disponible de MSR
- NcFTP (BSD) – desde el sitio "proyecto KAME"

Games

- Quakeforge – un puerto FreeBSD por Viagenie

IPsec

- IPv6 FreeS/WAN for Linux – desarrollado por 6INIT
- IPv6 Ipsec in KAME – KAME IPv6 soporta IPsec con Racoon

Java

- IPv6 Java for Windows – nota este no es un producto de Sun Javasoft
- Sun JDK – JAVA (TM)2, edición 1.4.1 FCS incluye soporte IPv6

Mail

- Exim – incluye soporte para IPv6
- Qmail – v1.03 parche por Kazunori Fujiwara
- Public Sendmail – versión 8.10 oficialmente soporta IPv6
- WIDE Sendmail – versión 8.9.1 de KAME
- Fetch mail – soporta IPv6 e IPsec

Mobile IPv6

- MIPL Mobile IPv6 for Linux – desarrollado por HUT, Finlandia y disponible libremente bajo GPL

Monitoring tools

- ASPath_tree – una herramienta para monitorear ruteo BGP4+
- COLD – un paquete sniffer de IPv6-aware

News

- INNv2.2.2 – IPv6 parche del sitio del norte japonés
- mnews – cliente de noticias para IPv6

Patch sites
- IPv6 Meat – parches para linux
- IPv6 patches – mantenido por Hajimu Umemoto
- KAME patches – del sitio japonés proyecto KAME
- KAME patch list – una lista de parches para muchas aplicaciones
- Linux IPv6 apps – del grupo de usuarios japonés de Linux
- Linux IPv6 apps list – por Peter Bieringer
- University of tromso – parches de el sitio Norwegian
- WIDE patches – del sitio WIDE del proyecto japonés
- Zama Networks – parches para una variedad de aplicaciones

Socket software
- IPv6 socket 1.1 – por la Universidad de Tromso

Vídeo y conferencia
- ISABEL – Grupo de trabajo y sistema de conferencia
- mpeg4ip – './bootstrap –enable-ipv6'
- Vic and Rat - herramientas UCL MICE
- Vic/Rat for WinXP – información de video/audio multicast por Microsoft

Web servers and clients
- Apache (linux) – del grupo de usuarios japonés Linux
- Apache (BSD) – del proyecto KAME
- Apache + mod_ssl – parches del Zama Networks
- Apache 2.0 –actualmente código beta, pero soporta IPv6
- Fnord! – un servidor de web para windows de MSR
- Lynx – puerto del texto basado en browser por Tromso
- Mini hhtpd – un servidor de web con soporte IPv6
- Mozilla – puerto del browser por KAME
- Thhtpd – un servidor de web que soporta IPv6
- W3m – un texto basado en browser que soporta IPv6

3.6.1 Implementaciones

Implementaciones en sistemas operativos

APPLE
- Jaguar – la versión MacOS X v10.2 tiene un stack de producción IPv6 y soporte de IPsec

BSD

- FreeBSD4.0 – incluye el stack KAME IPv6
- KAME – las diferentes versiones BSD estan unidas aquí
- INRIA – el desarrollo parece haber cedido para KAME
- NRL´s IPv6 - como distribuido de MIT (v7. 1 Dec ´98)+
- IPv6-DRET – Una implementación francesa.

COMPAQ

- Compaq IPv6 Information – acerca de IPv6 e implementaciones tru64/Open VMS

FPT/Netmanage

- OnNet Host Suite – soporte IPv6 para Windows 95/98/NT

Future Software

- FutureIPv6 Host –Código fuente portable en el producto Future Software Limited

Hitachi

- Toolnet6 – proporciona conectividad IPv6 para Pc Windows

HP

- HP-UX 11! – liberación de IPv6 (Agosto 2001)

IBM

- AIX 4.3 - Soporte IPv6 para el RS6000
- Next Generation Internet – IPv6 y muchos más
- OS/390 – un prototipo de trabajo

Integrate systems Inc. (ISI)

- Ipv6 in embedded systems – la primera compañía para lograr esto

Linux

- IPv6 users Group Up – sitio japonés pero la gran cantidad en Inglés
- IPv6 How to – por Peter Bieringer
- USAGI Proyect – Campo de juegos universales para IPv6 con WIDE, KAME Y TAHI
- Ipv6 Meat – parches para Linux
- Debian IPv6 Project – IPv6 para Debian Linux
- Linux IPv6 RPM Project – IPv6 en paquetes RPM

Microsoft

- Microsoft and IPv6 – Información general del producto
- Windows XP IPv6 FAQ - XP soporta IPv6

- Microsoft IPv6 Technology - for .NET, WinCE .NET and WinXP – mucha información al respecto
- Corona - con soporte IPv6
- Official Windows 2000 press release – información de IPv6 y Win 2000
- Windows 2000 preview version – disponible para Win2K SP1
- Microsoft Research and IPv6 - información general de MSR

Mentat
- Mentat TCP – TCP/IP stack incluyendo IPv6 y soporte IPv6sec

SCO
- UnixWare 7 – con soporte IPv6 API

SUN
- Solarios 8 – soporta navegación con IPv6

TRUMPET
- Winsock 5.0 – un stack IPv6 para Win9x/NT

Implementaciones en Routers

3 COM
- 6com.net – sitio de información 3Com sobre IPv6
- Technology info – Información general
- Enterprise OS software v11.4 – Manual de referencia con información de IPv6

6WIND
- IpV6 Edge Device – proporciona VPN, administración QoS e IPv4/v6 características de migración

CISCO
- Cisco IP Version 6 Solutions – información general.
- Configuring IOS Software - notas en 12.2 series.
- Cisco IOS Software Release - especificaciones y características para IPv6.
- IPv6 IOS - Cisco IOS para IPv6, con muchas ligas.
- 6NET - un proyecto Europeo coordinado por Cisco.

Ericsson Telebit
- IPv6 modules – ruteadores Telebit soporta comercialmente IPv6

Fujitsu
- GeoStream R900 Series – un producto con soporte IPv6.

Future Software

- FutureIPv6 Router – Código fuente portable en el producto de Future Software Limited

GateD Consortium
- GateD – El nuevo GateD 1.0 la "liberación" esta solo disponible a miembros del consorcio

Hitachi
- Hitachi Internetworking - soporta IPv6 en el producto GR2000
- NR60 Router – Ha soportado Ipv6 desde 1997

InternetShare
- All Aboard! Advanced Edition – IPv4/IPv6 ruteador basado en Linux

IP Infusion
- ZebOS Advanced Routing Suite - incluye funciones avanzadas de IPv6

Juniper
- JUNOS – sostendrá IPv6 para el 4° cuarto del 2001 (articulo japonés)

Multi-threaded Routing Toolkit (MRT)
- MRT-2 2.0a – desarrollado por la Universidad de Michigan
- MRT at Sourceforge – La última version de MRT esta aquí

Nortel Networks
- Nortel IPv6 technology – información y soporte IPv6

Zebra
- Zebra – el producto ruteador GNU soporta IPv6

Herramientas de transición

Translator software
- BT Ultima IPv6 access – Acceso IPv6/IPv6, NAT/PT y un tunnel
- Bump in the API – desde ETRI y software I2
- Socks5 IPv4-to-IPv6 translator – provee un servicio proxy
- NAT/PT translator – implementación espacio de usuario para Linux
- Windows IPv6/IPv4 translator – disponible desde MSR

Túnel software
- CSELT tunnel Broker - creador automático de tunnel IPv6
- Toolnet6 – proporciona una conectividad IPv6 para Windows PC
- V6tun – un BSD IPv6 una herramienta para hacer un túnel que puede ejecutarse a través de ssh

Web proxy/caché

- Squid – puerto del webcaché por KAME
- wwwofflev2.5 – un proxy para ver solamente sitios v6

Por último, como parte de las aplicaciones y tendencias en el futuro de IPv6, será como lo muestra la figura siguiente.

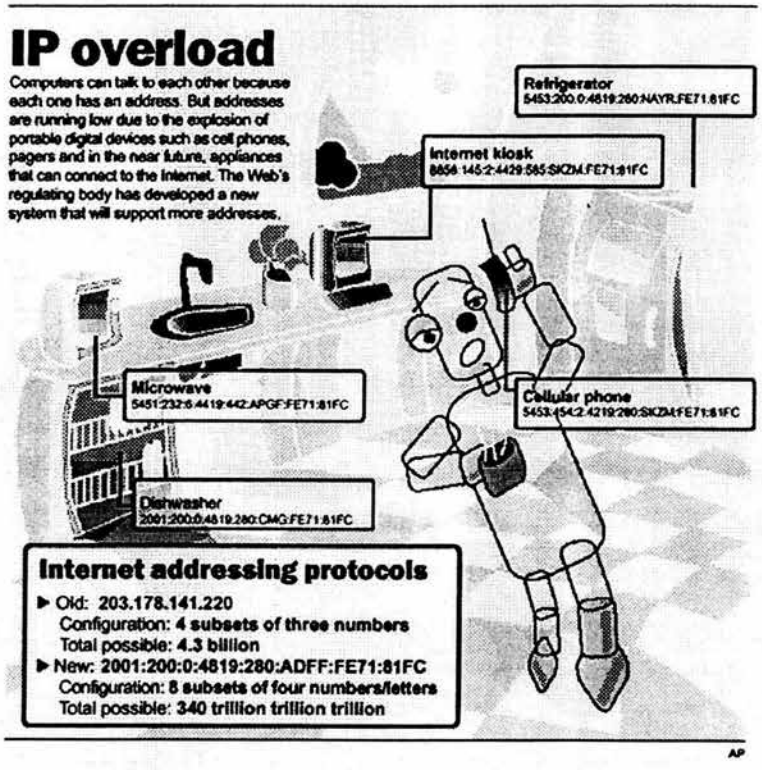
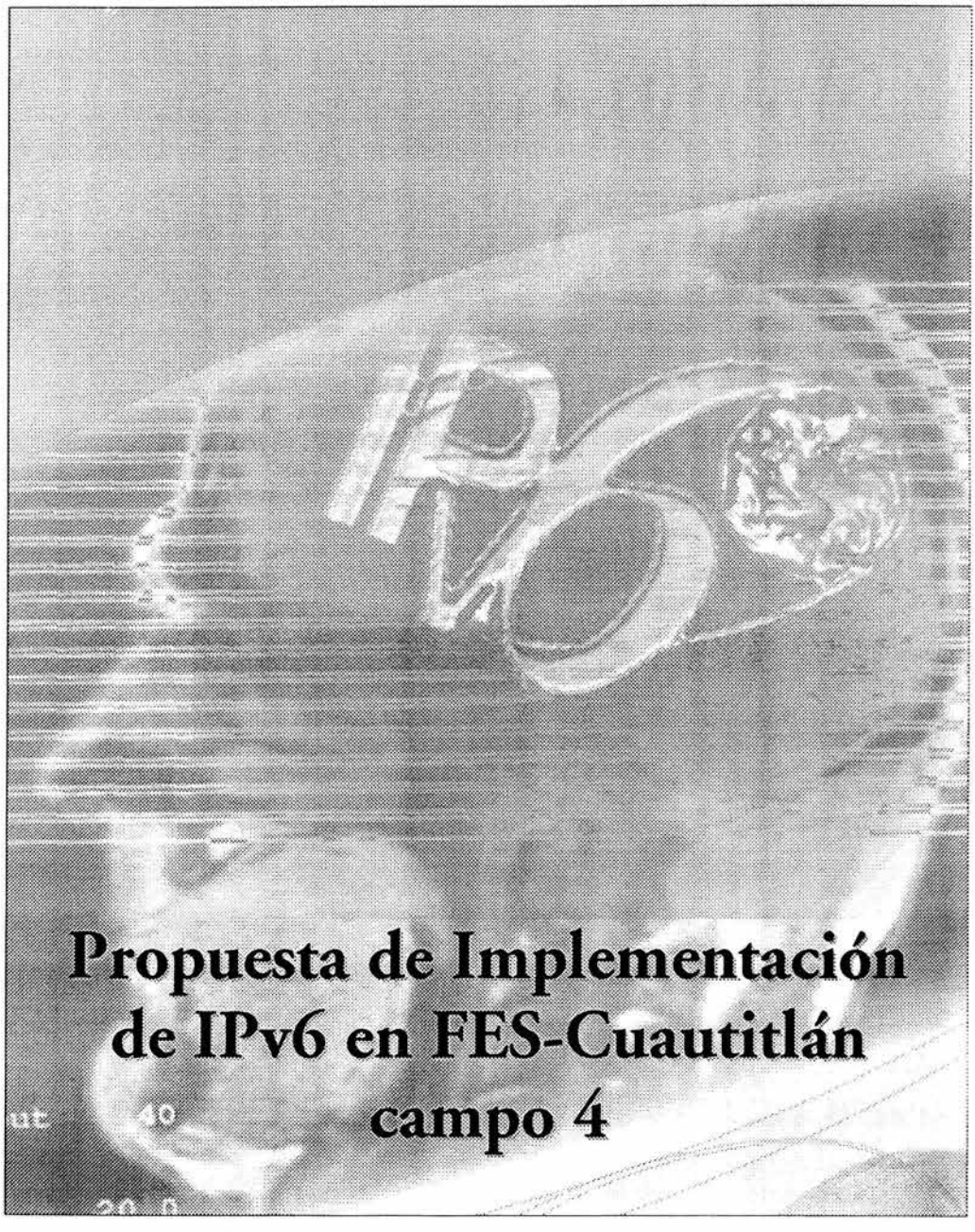


Figura 3.8 Aplicaciones de IPv6 en el futuro

Capítulo **4**



Propuesta de Implementación de IPv6 en FES-Cuautitlán campo 4

CAPITULO 4

Implementación de IPv6 en FES-Cuautitlán

Ya que IPv6 brinda mejores características que buscan resolver los problemas surgidos con el incremento del uso de las redes de computadoras, se implementará IPv6 en el campus Cuautitlán con fines experimentales y de investigación. Será la primera red IPv6 instalada en el campus

De aquí en adelante se hablará de los pasos necesarios para hacer la implementación de IPv6, utilizando los recursos con la que cuenta la FES-Cuautitlán, así mismo se demostrará la conexión con IPv6

4.1 Requisitos

En primera instancia se tiene que solicitar un bloque de direcciones IPv6 al grupo de trabajo de IPv6 de la UNAM. Se puede también solicitar a otra institución u organización un bloque de direcciones IPv6 que tenga la facultad de delegar este tipo de direcciones ó se puede hacer simplemente un túnel 6to4 pero debido a que Cuautitlán es parte de la UNAM se hará la solicitud con la misma.

Como un primer paso es solicitar un bloque de direcciones IPv6 de pruebas con el tiempo y una vez que se obtiene experiencia se puede solicitar un bloque de direcciones IPv6 de producción.

Hablando en particular de la FES-Cuautitlán (FES-C) se solicitará un bloque de direcciones IPv6 de pruebas. Para ello se tiene que enviar un correo solicitando dicho bloque a la dirección de correo electrónico:

staff_ipv6@ipv6.unam.mx

Hecho esto, se recibirá una respuesta con los pasos a seguir para la obtención del bloque solicitado.

Uno de los pasos es entrar a la página de IPv6 de la UNAM* aquí nos dirigimos a la sección de *Solicitud de bloques*, ahí se encuentra la solicitud que se tiene que llenar. Aquí nos pide ciertos datos como por ejemplo, nombre de la institución, contacto administrativo, contacto técnico, justificación, entre otras. Ésta forma sólo puede ser llenada por el **Administrador de la red**.

En la figura 4.1 se muestra el llenado de ésta solicitud para obtener un bloque de direcciones IPv6 para la FES- Cuautitlán.

* Para más detalles visitar la página http://www.ipv6.unam.mx

SU SOLICITUD HA SIDO REGISTRADA
CON LOS SIGUIENTES DATOS:

INSTITUCIÓN O EMPRESA:

Nombre Completo: **Universidad Nacional Autónoma de México**
Siglas: **UNAM**
Calle y No.: **Km. 2.5 Carr. Cuautitlán-Teoloyucan**
Colonia: **San Sebastián Xhala**
Del. o Municipio: **Cuautitlán Izcalli**
Ciudad: **Cuautitlán Izcalli**
Estado: **EDO**
País: **México**
Código Postal: **54714**

CONTACTO ADMINISTRATIVO:

NIC-Handle: **jmhd**
Título: **Ingeniero**
Nombre (s): **Jesús Moisés**
Apellido Paterno: **Hernández**
Apellido Materno: **Duarte**
Puesto: **Jefe del Centro de Cómputo**
E-mail: **hduarte@servidor.unam.mx**
Teléfono: **31860**
Fax: **31833**

CONTACTO TÉCNICO:

Misma persona que Contacto Administrativo

JUSTIFICACIÓN:

Planeación: **La infraestructura con la que cuenta actualmente la Facultad ha crecido en buena medida, lo que obliga a buscar utilizar nuevas tecnologías que permitan hacer llegar la conexión a Internet a más equipos.**
Dirección IPv4 del equipo remoto: **132.248.102.201**
Número de túneles a solicitar: **1**
Número de túneles asignados previamente: **0**
Plataforma (SistemaOperativo): **Linux SuSE 7.3**
Service Pack, Kernel o Parches:
Equipo:

Muy pronto recibirá un mensaje en la dirección de correo **hduarte@servidor.unam.mx** con información adicional.

Figura 4.1 Registro del bloque IPv6

Una vez que se lleno la solicitud, se recibirá nuevamente un correo pero ahora con los datos necesarios para crear un túnel y así mismo configurar el bloque solicitado.

A continuación pondré el texto del correo recibido del grupo de trabajo de IPv6 de la UNAM:

De acuerdo a la informacion que nos proporciono, ha sido aceptada la Solicitud de Bloque y Tunel con la sig. informacion:

 # Solicitud: 55

Datos a configurar en su equipo (ruteador):
Dir. IPv6: 3FFE:8070:1022::2/64
Dir. IPv4: 132.248.102.201

Datos del equipo de la UNAM:
Dir. IPv6: 3FFE:8070:1022::1/64
Dir. IPv4: 132.248.108.254
Tipo de Conexion: STATIC

Bloque pNLA delegado: 3FFE:8070:1022::/48 para FES-Cuatitlan

Para cualquier aclaracion y comentario, contestar este mensaje por favor con la informacion que considere pertinente.

NOTA: Una vez configurado el Tunel, favor de checar la conexion utilizando herramientas como Ping y Traceroute a la dir: 3FFE:8070:1022::1 y otras de IPv6, enviarnos sus resultados.

Asi mismo, consultar por favor la sig. direccion de Internet para registrar su Sitio (Bloque IPv6) en 6Bone:
www.ipv6.unam.mx/6bone.html

Por su atencion, gracias
SALUDOS[*]

Como parte de los requisitos se debe contar con el material necesario para hacer la instalación del stack IPv6, los cuales serán:

Material

- 3 PCs - pentium o cualquiera de las familias de Intel o AMD, con un mínimo de 128 Mb de memoria RAM y de disco duro con un de mínimo 3 Gb.
- 3 Tarjetas de red –10/100 Ethernet cualquier marca
- 3 Patch cord – UTP cat. 5
- 3 nodos de red – cada uno con una dirección IPv4 fija y válida

[*] sic

Software

- Linux – cualquier distribución de preferencia Red Hat ó Linux Mandrake última versión liberada
- Windows – Cualquier familia de Windows de preferencia Windows2000 en inglés o WindowsXP
- BSD –Cualquiera de su distribución (FreeBSD, openBSD, netBSD), liberada.

Hablando del software, para la maquina principal se utilizará Linux para configurar el túnel y a su vez se ocupará como ruteador de paquetes IPv6. En las otras máquinas se puede instalar el sistema operativo OpenBSD y en la última máquina se instalará cualquier sistema operativo de la Familia de Windows en este caso será WindowsXP.

Con esta variedad de sistemas operativos se pretende comprobar que efectivamente se puede habilitar IPv6 en cada uno de ellos, claro con sus propias limitaciones, recordemos que es un proyecto y que está en desarrollo por lo que cada versión nueva de cada sistema operativo tendrá que estar mejor soportado el stack de IPv6.

En resumen, los pasos a seguir son:

- Activar el stack de IPv6 en los equipos.
- Crear el túnel hacia el equipo extremo.
- Preparar el router para que soporte IPv6.
- Activar protocolos de ruteo (RIPng y BGP).
- Agregar otros equipos (clientes) a la isla IPv6 de la FES-Cuautitlán.

4.2 Esquema para la implementación de IPv6

Continuando con la propuesta de implementación en la FES-Cuautitlán se plantea un esquema para la implementación de IPv6, este esquema está ilustrado en la figura 4.2

El router que tiene la UNAM soporta "doble pila" y también soporta IPv6 nativo, por lo que, de nuestro lado se tiene que configurar una PC que haga la función de ruteador exclusivamente para IPv6, porque el *router* con el que cuenta la FES-Cuautitlán no soporta IPv6 por las características propias del equipo y sobre todo que no soporta las nuevas tecnologías que han surgido recientemente, es por eso que se tiene que hacer el arreglo como se muestra en la figura 4.2.

Tenemos dos routers, uno que pertenece al grupo de trabajo de IPv6 etiquetado como UNAM, en el otro extremo se tiene el router de Cuautitlán Campo 4, posteriormente se tiene una PC que contendrá el túnel IPv6sobreIPv4 y a la vez será configurado para que sea ruteador de paquetes IPv6. "Detrás" de éste, estarán todos los clientes que quieran o requieran una conexión IPv6.

ESQUEMA DE LA PROPUESTA DE IMPLEMENTACIÓN DE IPv6 EN LA FES-C

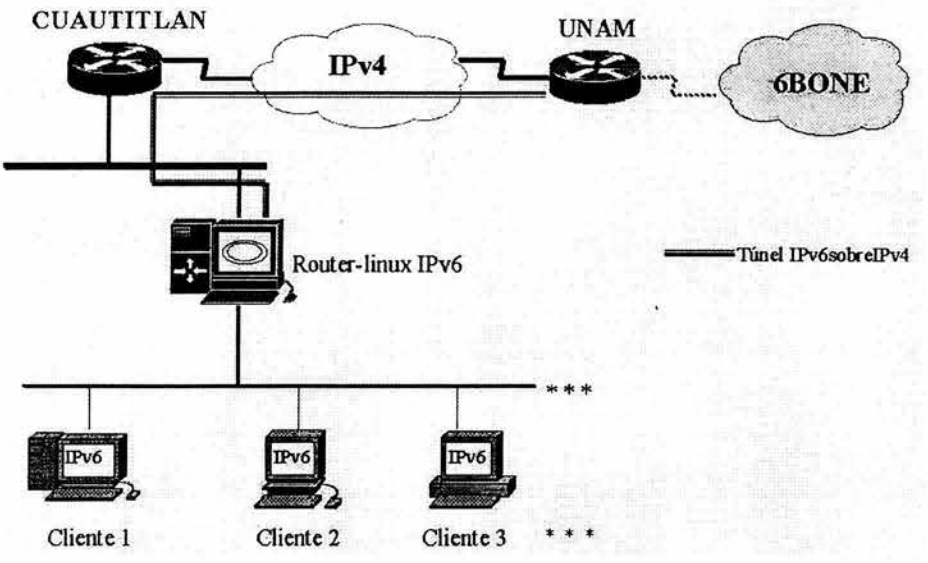


Figura 4.2 Esquema de Implementación de IPv6 en FES-C campo 4

4.3 Esquema de direccionamiento IPv6

Ahora bien para que se pueda tener una conexión con IPv6 se necesita un esquema de direccionamiento, una base de dónde partir, y así poder asignar direcciones IPv6 a los clientes o usuarios que estén interesados en tener una conexión con IPv6. Para ello se propone un esquema de direccionamiento IPv6 en la FES Cuautitlán y lo planteé de la siguiente manera:

a) Esta dirección 3ffe:8070:1022:1::/64 será para una subred derivado del bloque delegado para la FES-C

El propósito de poner el prefijo /64 es para ver el comportamiento de la autoconfiguración que se le asignará a cada cliente. Según vaya creciendo la red IPv6, las nuevas tendencias y requerimientos que la FES-C tenga en un futuro. Se irá modificando paulatinamente este esquema para tener un mejor aprovechamiento de las direcciones y una mejor distribución de las mismas.

Cómo una primera propuesta se tiene el planteamiento de un esquema de direccionamiento IPv6 que se ilustra en la figura 4.3

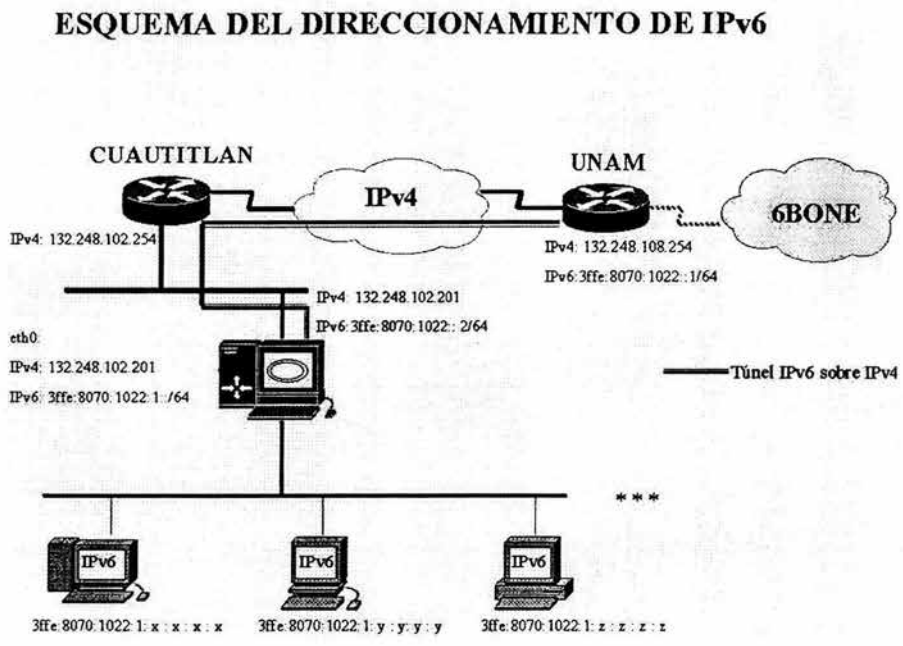


Figura 4.3 Esquema de direccionamiento IPv6 para la FES-C

4.4 Habilitación de IPv6 en los equipos

Una vez que se hizo la petición del bloque se procede, según requisitos, a configurar el túnel. Para ello se necesita configurar una PC la cual tendrá el túnel IPv6 sobre IPv4 y esta misma será PC utilizada como router-IPv6

Primero se tiene que cargar el modulo IPv6 de forma manual, posteriormente se hará un arreglo para que cuando se inicia la maquina lo cargue automáticamente.

4.4.1 Configuración General

Continuando ahora con la configuración de los equipos, se realizó la configuración en una PC con Linux Mandrake 9.1 con una versión de kernel 2.4.21-0.13mdk, que trabajará principalmente como router de IPv6 y se configuraron dos PCs como clientes, uno usando WindowsXP y el otro openBSD. Enseguida se presentarán las configuraciones realizadas en los equipos.

Configuración del túnel

El túnel de IPv6 sobre IPv4 se montará en la PC que tiene Linux Mandrake 9.1 que estará en FES-Cuautitlán en el otro extremo del túnel estará conectado a un router Cisco 2600. Los datos de la configuración son:

Datos a configurar en el equipo de la FES-C

IPv4: 132.248.102.201
IPv6: 3FFE:8070:1022::2/64

Datos del equipo de la UNAM:

IPv4: 132.248.108.254
IPv6: 3FFE:8070:1022::1/64

Con estos valores se realiza la configuración del túnel.

Antes que cualquier cosa, se necesita cargar el módulo *ipv6*, para esto existen varias formas de hacerlo, primero se hará de forma manual*.

1. Se carga el módulo IPv6 de la siguiente manera:

[root@ipv6 greg]# modprobe ipv6

2. Usando los datos para crear el túnel se realiza la configuración. Hacemos el siguiente proceso:

[root@ipv6 greg]# ip tunnel add sit1 mode sit remote 132.248.108.254 local 132.248.102.201 ttl 64

3. Se levanta la interfaz

[root@ipv6 greg]# ip link set sit1 up

4. Se revisan las interfaces

```
[root@ipv6 greg]# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:01:80:2A:61:CA
          inet addr:132.248.102.201  Bcast:132.248.102.255  Mask:255.255.255.0
          inet6 addr: fe80::201:80ff:fe2a:61ca/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7017 errors:0 dropped:0 overruns:0 frame:0
          TX packets:163 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
```

* se debe tener privilegios de root para hacer cualquier cambio en el sistema operativo)

```
        RX bytes:455789 (445.1 Kb)  TX bytes:10576 (10.3 Kb)
        Interrupt:20 Base address:0xa000

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:104 errors:0 dropped:0 overruns:0 frame:0
        TX packets:104 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:7272 (7.1 Kb)  TX bytes:7272 (7.1 Kb)

sit0    Link encap:IPv6-in-IPv4
        NOARP  MTU:1480  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

sit1    Link encap:IPv6-in-IPv4
        inet6 addr: fe80::84f8:66c9/128 Scope:Link
        UP POINTOPOINT RUNNING NOARP  MTU:1480  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

5. Se agrega la dirección IPv6 correspondiente al túnel

```
[root@ipv6 greg]# ip address add 3FFE:8070:1022::2/64 dev sit1
```

6. Se crea una ruta estática

```
[root@ipv6 greg]# route -A inet6 add ::0/ gw 3FFE:8070:1022::1 dev sit1
```

7. Se revisan las interfaces nuevamente

```
[root@ipv6 greg]# ifconfig -a
eth0    Link encap:Ethernet  HWaddr 00:01:80:2A:61:CA
        inet addr:132.248.102.201  Bcast:132.248.102.255  Mask:255.255.255.0
        inet6 addr: 3ffe:8070:1022:1::/64 Scope:Global
        inet6 addr: fe80::201:80ff:fe2a:61ca/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:2151872 errors:0 dropped:0 overruns:0 frame:0
        TX packets:30469 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:100
        RX bytes:151378 (144.3 kb)  TX bytes:5385 (5.1 kb)
        Interrupt:20 Base address:0xa000
```

```
lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:4310 errors:0 dropped:0 overruns:0 frame:0
        TX packets:4310 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:264 (25.8 Kb)  TX bytes:26 (2.8 Kb)

sit0    Link encap:IPv6-in-IPv4
        inet6 addr: ::127.0.0.1/96 Scope:Unknown
        inet6 addr: ::132.248.102.201/96 Scope:Compat
        UP RUNNING NOARP  MTU:1480  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

sit1    Link encap:IPv6-in-IPv4
        inet6 addr: 3ffe:8070:1022::2/64 Scope:Global
        inet6 addr: fe80::84f8:66c9/128 Scope:Link
        UP POINTOPOINT RUNNING NOARP  MTU:1480  Metric:1
        RX packets:2620 errors:0 dropped:0 overruns:0 frame:0
        TX packets:2818 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:0 (0.0 Kb)  TX bytes:0 (0.0 Kb)
```

Con esta configuración ya se tiene el túnel por el cual "saldrán" los paquetes de IPv6 a Internet. Para esto se tiene que comprobar usando los siguiente comandos:

8. Con el comando "ping6" comprobamos la conectividad con el otro extremo del túnel

```
[root@ipv6 greg]# ping6 3FFE:8070:1022::1
PING 3FFE:8070:1022::1(3ffe:8070:1022::1) 56 data bytes
64 bytes from 3ffe:8070:1022::1: icmp_seq=1 ttl=64 time=8.29 ms
64 bytes from 3ffe:8070:1022::1: icmp_seq=2 ttl=64 time=7.68 ms
64 bytes from 3ffe:8070:1022::1: icmp_seq=3 ttl=64 time=7.14 ms
64 bytes from 3ffe:8070:1022::1: icmp_seq=4 ttl=64 time=7.35 ms
64 bytes from 3ffe:8070:1022::1: icmp_seq=5 ttl=64 time=7.57 ms
64 bytes from 3ffe:8070:1022::1: icmp_seq=6 ttl=64 time=7.27 ms
64 bytes from 3ffe:8070:1022::1: icmp_seq=7 ttl=64 time=8.13 ms

--- 3FFE:8070:1022::1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6061ms
rtt min/avg/max/mdev = 7.140/7.636/8.295/0.420 ms
```

9. Con el comando "traceroute6" se traza una ruta hacia el otro extremo del túnel

[root@ipv6 greg]# traceroute6 3FFE:8070:1022::1
traceroute to 3FFE:8070:1022::1 (3ffe:8070:1022::1) from 3ffe:8070:1022::2, 30 hops max, 16 byte packets
*1 3ffe:8070:1022::1 (3ffe:8070:1022::1) 47.129 ms * 7.055 ms*

Como se puede observar sólo hay un paso para llegar a nuestro destino, que es el otro extremo del túnel.

Existen otras formas de configurar un túnel en Linux* depende de la distribución, la versión de Linux y la sobre todo la versión del Kernel.

Modificando archivos para inicio automático

Por último para no tener que hacer el proceso de configuración de túneles cada vez que se reinicie la máquina ó se desconfigure el túnel por alguna razón; se tienen que configurar algunos archivos para que se cargue en el sistema en forma automática, además que este método es el más recomendado. Sólo se tienen que editar los siguientes archivos*.

1. Comenzamos con el archivo que tendrá la configuración de la red:

*/etc/sysconfig/**network***

HOSTNAME=ipv6
NETWORKING=yes
GATEWAY=132.248.102.254
NETWORKING_IPV6=yes #habilita el inicio con IPv6
IPV6FORWARDING=yes #habilita el avance de paquetes IPv6
IPV6AUTOCONF=yes #Habilita la autoconfiguración IPv6
IPV6_DEFAULTDEV=sit1 #especifica la interfaz de salida
IPV6_AUTOTUNNEL=yes #habilita automáticamente un túnel IPv6
GATEWAYDEV=eth0

2. Después el archivo donde se encuentra la configuración de la interfaz de red:

*/etc/sysconfig/network-scripts/**ifcfg-eth0***

DEVICE=eth0
BOOTPROTO=static
IPADDR=132.248.102.201
NETMASK=255.255.255.0
NETWORK=132.248.102.0
BROADCAST=132.248.102.255
ONBOOT=yes
MII_NOT_SUPPORTED=yes

---

* Para mayor información consultar la pagina http://www.tldp.org/HOWTO/Linux+IPv6-HOWTO/

* se resaltarán con negro las líneas que deberán ser agregadas

```
IPV6INIT=yes                          #habilita el inicio de IPv6
IPV6ADDR=3FFE:8070:1022:1::1/64       #habilita la dirección IPv6
```

3. Por último el archivo que contendrá el túnel:

*/etc/sysconfig/network-scripts/**ifcfg-sit1***

```
DEVICE=sit1                           #especifica el nombre de la interfaz
BOOTPROTO=none
ONBOOT=yes
IPV6INIT=yes                          #habilita el inicio de IPv6
IPV6TUNNELIPV4=132.248.108.254        #dirección IPv4 remoto del túnel
IPV6TUNNELLOCAL=132.248.102.201       #dirección IPv4 local del túnel
IPV6ADDR=3FFE:8070:1022::2/64         #habilita la dirección IPv6
IPV6_DEFAULTGW=3FFE:8070:1022::1      #hacia donde será ruteado
```

Sólo falta configurar el archivo por el cual se hará la autoconfiguración de los clientes. El demonio radvd se configuró de la siguiente manera:

/etc/radvd.conf

```
interface eth0
{
    AdvSendAdvert on;
    MinRtrAdvInterval 3;
    MaxRtrAdvInterval 10;
    prefix 3FFE:8070:1022:1::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr on;
    };

};
```

Una vez que se tienen los archivos anteriores correctamente configurados, reiniciamos la máquina para que tome los valores respectivos y se confirma la configuración de las interfaces:

```
[root@ipv6 greg]# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:01:80:2A:61:CA
          inet addr:132.248.102.201  Bcast:132.248.102.255  Mask:255.255.255.0
          inet6 addr: 3ffe:8070:1022:1::/64 Scope:Global
          inet6 addr: fe80::201:80ff:fe2a:61ca/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2151872 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30469 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:151378117 (144.3 Mb)  TX bytes:5385451 (5.1 Mb)
```

```
          Interrupt:20 Base address:0xa000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:4310 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4310 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:264074 (257.8 Kb)  TX bytes:264074 (257.8 Kb)

sit0      Link encap:IPv6-in-IPv4
          inet6 addr: ::127.0.0.1/96 Scope:Unknown
          inet6 addr: ::132.248.102.201/96 Scope:Compat
          inet6 addr: ::132.248.249.2/96 Scope:Compat
          UP RUNNING NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

sit1      Link encap:IPv6-in-IPv4
          inet6 addr: 3ffe:8070:1022::2/64 Scope:Global
          inet6 addr: fe80::84f8:66c9/64 Scope:Link
          inet6 addr: fe80::84f8:f902/64 Scope:Link
          UP POINTOPOINT RUNNING NOARP  MTU:1480  Metric:1
          RX packets:2620 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2818 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:419533 (409.7 Kb)  TX bytes:321223 (313.6 Kb)
```

Se puede observar que efectivamente se encuentra configurado la red IPv6 tal como se planteo.

Configuración del router-IPv6

El siguiente paso es instalar un software que tenga la función de router para intercambiar rutas con nuestro proveedor y el que use el *router discovery** para que las pilas de IPv6 se puedan autoconfigurar.

Para la configuración del router se utilizó el software "*Quagga*" para Linux, mejor conocido como Zebra éste software se puede descargar de la página: http://www.quagga.org

Una vez que se descargo y se instaló el software para ruteo, se continúa con la configuración de éste para que los equipos sean capaces de autoconfigurarse y al

* Vid. capítulo 2, apartado 2.4

mismo tiempo que se establezca una sesión con BGP4+ con el router remoto, para ello necesitan básicamente dos archivos de configuración: zebra.conf y bgpd.conf ambos ubicados en /usr/local/etc

Se configuro el archivo *zebra.conf* de la siguiente manera:

```
!
hostname zebra
password unam-fesc
enable password fesc
!
! Interface's description.
!
interface eth0
  ipv6 nd send-ra
  ipv6 nd prefix-advertisement 3ffe:8070:1022:1::/64
!
interface lo
!
interface sit0
      ipv6 nd suppress-ra
!
interface sit1
      ipv6 nd suppress-ra
!
!
no ip forwarding
!
line vty
!
end
!
log file zebra.log
```

Interesan el *hostname*, las claves y las líneas que comienzan con *ipv6*. Mediante la línea *ipv6 nd send-ra* con esto se le dice a Zebra que haga anuncio de router y con *ipv6 nd prefix-advertisement* se especifica la red y el prefijo a anunciar. Esta configuración se logró creando el archivo en forma manual mediante un editor y luego se inicia Zebra de la siguiente forma:

```
[root@ipv6 greg]# service zebra start

[root@ipv6 greg]# /usr/local/zebra/zebra -d
```

Lo que a continuación se hace es configurar el BGP éste protocolo se maneja en base a números de sistemas autónomos (ASN); en el caso particular de la FES-C, es el mismo sistema autónomo que tiene la UNAM (ASN 278). Hay que tener

cuidado con esto y no elegir cualquier número como ASN ya que podríamos generar problemas graves de ruteo.

Se llama sistema autónomo a `"la red de alguien", esto es, a una red o conjunto de redes controladas por una única entidad administrativa. Cada sistema autónomo decide cómo manejar sus redes internas independientemente de cualquier otro, y se conecta a otros sistemas autónomos por medio de protocolos de ruteo externos.

Basicamente, lo que hace BGP* es establecer una relación de pares entre dos routers que pertenecen a distintos sistemas autónomos y esos routers se informan entre sí acerca de a qué redes conocen como hacer llegar paquetes, con lo que pueden formarse los mapas.

De esta manera se configura el archivo bgpd.conf y queda de la siguiente manera:

```
hostname bgpd
password unam-fesc
enable password fesc
log stdout
!
bgp config-type cisco
!
router bgp 278
 bgp router-id 132.248.102.201
 neighbor 3ffe:8070:1022::1 remote-as 278
 neighbor 3ffe:8070:1022::1 description tunel ipv6 con DGSCA
¡ no auto-summary
!
 address-family ipv6
 network 3ffe:8070:1022::/48
 neighbor 3ffe:8070:1022::1 activate
 neighbor 3ffe:8070:1022::1 send-community both
 neighbor 3ffe:8070:1022::1 soft-reconfiguration inbound
exit-address-family
!
line vty
!
log file /var/log/zebra/bgpd.log
```

Aquí se aprecian las configuraciones de las redes involucradas en el ruteo, primero se define la red local, luego se da el extremo remoto del túnel asociado a ciertos parámetros: AS remoto, interfaz, una descripción, entre otras.

Para asegurarse se podría reiniciar el Zebra de la siguiente manera:

---

* Véase capítulo 2.9.3

[root@ipv6 greg]# Service zebra restart

[root@ipv6 greg]# /usr/local/sbin/bgpd -d

Después iniciar el bgpd y confirmar la configuración que se realizo.

La forma de confirmar la configuración es haciendo un telnet al puerto 2601 de la siguiente manera:

[root@ipv6 greg]#telnet localhost 2601
Trying 127.0.0.1...
Connected to localhost (127.0.0.1).
Escape character is '^]'.

Hello, this is quagga (version 0.96.4).
Copyright 1996-2002 Kunihiro Ishiguro.

User Access Verification

Password:
ipv6> en
Password:
ipv6#

ipv6# sh run

Current configuration:
!
hostname ipv6
password unam-fesc
enable password fesc
log file /var/log/zebra.log
!
interface eth0
no ipv6 nd suppress-ra
ipv6 nd prefix-advertisement 3ffe:8070:1022:1::/64 2592000 604800 onlink autoconfig
!
interface lo
!
interface sit0
ipv6 nd suppress-ra
!
interface sit1
ipv6 nd suppress-ra
!
no ip forwarding
!

```
line vty
!
end

ipv6# exit
Connection closed by foreign host.
```

Aquí se muestra la configuración de acuerdo a lo ingresado en el archivo zebra.conf. A continuación se presenta la configuración de bgpd, por el puerto 2605:

```
[root@ipv6 greg]# telnet localhost 2605
Trying 127.0.0.1...
Connected to localhost (127.0.0.1).
Escape character is '^]'.

Hello, this is quagga (version 0.96.4).
Copyright 1996-2002 Kunihiro Ishiguro.


User Access Verification

Password:
bgpd> en
Password:
bgpd#

bgpd# sh run

Current configuration:
!
hostname bgpd
password unam-fesc
enable password fesc
log stdout
!
router bgp 278
 bgp router-id 132.248.102.201
 neighbor 3ffe:8070:1022::1 remote-as 278
 neighbor 3ffe:8070:1022::1 description tunel ipv6 con DGSCA
!
 address-family ipv6
 network 3ffe:8070:1022::/48
 neighbor 3ffe:8070:1022::1 activate
 neighbor 3ffe:8070:1022::1 soft-reconfiguration inbound
 exit-address-family
!
line vty
!
```

```
end

bgpd# sh ip bgp summ
BGP router identifier 132.248.102.201, local AS number 278
1 BGP AS-PATH entries
0 BGP community entries

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
3ffe:8070:1022::1
            4   278       0       0        0    0    0 00:01:18 Active

Total number of neighbors 1
bgpd#

bgpd# exit
Connection closed by foreign host.
```

Con esto basta para que se realice el trabajo de ruteo, ahora sólo se necesita agregar clientes a esta red. Una vez hecho esto y después de innumerables pruebas del sistema, se ha observado que **el anuncio de router es válido** solamente **si el prefijo es de 64 bits**, ya que la MAC de la tarjeta de red siempre es usada para configurar la dirección IPv6. Esto quiere decir, que se podría usar este método siempre y cuando se cuente con un prefijo menor o igual a 64 bits, de otra modo se debe configurar a mano en cada equipo.

4.4.2 Configuración de los clientes

Es el momento de configurar a los clientes, cada PC una tiene un sistema operativo diferente y por lo cual su configuración será diferente, continuemos con la configuración de estos equipos.

Configuración de un túnel en WindowsXP

Procedimiento para habilitar el stack IPv6

1. Entrar a sesión como administrador ó como usuario con permisos de administración.
2. Abrir una ventana de línea de comandos.
3. Teclear el siguiente comando:

ipv6 install

Debe obtener lo siguiente:

C:\>ipv6 install
Instalando...
Con éxito.

Se configura la red a la que pertenece, el gateway y además que anuncie dicha red.

C:\> ipv6 ifc 4 forwards adv
C:\> ipv6 rtu 3ffe:8070:1022:1:: /64 4
C:\> ipv6 rtu ::/0 4/3ffe:8070:1022::1

Finalmente se establece el router por defecto a través de la interfaz correspondiente, en este caso 4.

C:\> ipv6 rtu ::/0 4/3ffe:8070:1022::1

Comprobación del funcionamiento del stack IPv6:

1. Ahora se comprueba que se ha habilitado el stack de IPv6, tecleando el siguiente comando

C:\>ipv6 if ***(muestra información de por lo menos 4 interfaces lógicas)***

Interfaz 4: Ethernet: Conexión de área local
usa unidad de detección de equipos cercanos (Neighbor Discovery)
utiliza descubrimiento de enrutador
dirección de capa de vínculo: 00-90-27-c7-0c-12
preferred global 3ffe:8070:1022:1:8496:58a2:629a:f2d0, duración 6d23h51m23s/ 23h49m1s (anónimo)
preferred global 3ffe:8070:1022:1:290:27ff:fec7:c12, duración 29d23h54m8s/6d 23h54m8s (público)
preferred link-local fe80::290:27ff:fec7:c12, duración infinite
multidifusión interface-local ff01::1, 1 referencias, no se puede informar
multidifusión link-local ff02::1, 1 referencias, no se puede informar
multidifusión link-local ff02::1:ffc7:c12, 3 referencias, último informe
multidifusión link-local ff02::1:ff9a:f2d0, 2 referencias, último informe
vínculo MTU 1500 (vínculo MTU verdadero 1500)
limite de saltos actual 64
tiempo accesible 44500ms (base 30000ms)
intervalo de retransmisión 1000ms
transmisiones DAD 1
Interfaz 3: Seudo interfaz de túnel 6to4
no usa unidad de detección de equipos cercanos (Neighbor Discovery)
no utiliza decubrimiento de enrutador
preferred global 2002:84f8:f969::84f8:f969, duración infinite

vínculo MTU 1280 (vínculo MTU verdadero 65515)
limite de saltos actual 128
tiempo accesible 29000ms (base 30000ms)
intervalo de retransmisión 1000ms
transmisiones DAD 0
Interfaz 2: Seudo interfaz de túnel automático
no usa unidad de detección de equipos cercanos (Neighbor Discovery)
no utiliza decubrimiento de enrutador
Dirección de capa de enlace del enrutador: 0.0.0.0
Dirección IPv4 con EUI-64 incrustado: 0.0.0.0
preferred link-local fe80::5efe:132.248.249.105, duración infinite
preferred global ::132.248.249.105, duración infinite
vínculo MTU 1280 (vínculo MTU verdadero 65515)
limite de saltos actual 128
tiempo accesible 32000ms (base 30000ms)
intervalo de retransmisión 1000ms
transmisiones DAD 0
Interfaz 1: Seudo interfaz de bucle invertido
no usa unidad de detección de equipos cercanos (Neighbor Discovery)
no utiliza decubrimiento de enrutador
dirección de capa de vínculo:
preferred link-local ::1, duración infinite
preferred link-local fe80::1, duración infinite
vínculo MTU 1500 (vínculo MTU verdadero 4294967295)
limite de saltos actual 128
tiempo accesible 23500ms (base 30000ms)
intervalo de retransmisión 1000ms
transmisiones DAD 0

2. Para comprobar la interconexión de las interfaces virtuales, ejecutar el comando "ping6" a cada una de las direcciones IPv6 que aparecen en cada interfaz:

C:\>ping6 ::1

Haciendo ping ::1
de ::1 con 32 bytes de datos:

Respuesta desde ::1: bytes=32 tiempo<1m
Respuesta desde ::1: bytes=32 tiempo<1m
Respuesta desde ::1: bytes=32 tiempo<1m
Respuesta desde ::1: bytes=32 tiempo<1m

Estadísticas de ping para ::1:
Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos)
Tiempos aproximados de ida y vuelta en milisegundos:
Mínimo = 0ms, Máximo = 0ms, Media = 0ms

Al hacerlo de esta manera se pierde la configuración cuando se reinicie el equipo. La forma de hacer permanente esta configuración es crear un archivo de extensión *.cmd* que contenga las instrucciones vistas anteriormente y posteriormente añadirlo en el Programador de Tareas para que sea ejecutado cada vez que se inicia el equipo.

Configuración de un túnel en openBSD 3.3

Se checa primero el estado de las interfaces de red

```
obsd-ipv6# ifconfig -a
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33224
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x5
lo1: flags=8008<LOOPBACK,MULTICAST> mtu 33224
fxp0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        address: 00:50:8b:01:cf:87
        media: Ethernet autoselect (100baseTX full-duplex)
        status: active
        inet 132.248.102.225 netmask 0xffffff00 broadcast 132.248.102.255
        inet6 fe80::250:8bff:fe01:cf87%fxp0 prefixlen 64 scopeid 0x1
pflog0: flags=0<> mtu 33224
pfsync0: flags=0<> mtu 2020
sl0: flags=c010<POINTOPOINT,LINK2,MULTICAST> mtu 296
sl1: flags=c010<POINTOPOINT,LINK2,MULTICAST> mtu 296
ppp0: flags=8010<POINTOPOINT,MULTICAST> mtu 1500
ppp1: flags=8010<POINTOPOINT,MULTICAST> mtu 1500
tun0: flags=10<POINTOPOINT> mtu 3000
tun1: flags=10<POINTOPOINT> mtu 3000
enc0: flags=0<> mtu 1536
bridge0: flags=0<> mtu 1500
bridge1: flags=0<> mtu 1500
vlan0: flags=0<> mtu 1500
        address: 00:00:00:00:00:00
vlan1: flags=0<> mtu 1500
        address: 00:00:00:00:00:00
gre0: flags=9010<POINTOPOINT,LINK0,MULTICAST> mtu 1450
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
gif1: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
gif2: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
gif3: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
```

Sólo a manera de ejemplo se hará un túnel en esta PC

Creación del túnel ipv6 en openbsd

```
bsd-ipv6# ifconfig gif0 giftunnel 132.248.102.225 132.248.102.201
```

```
bsd-ipv6# ifconfig gif0 inet6 3ffe:8070:1022:1::1

bsd-ipv6# route add -inet6 3ffe:8070:1022:1::1 -prefixlen 64 3ffe:8070:1022::4

bsd-ipv6# route add -inet6 default 3ffe:8070:1022::1

bsd-ipv6# route6d
```

Ahora se vuelve a checar las interfaces y así podemos ver los cambios

```
obsd-ipv6# ifconfig -a
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33224
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x5
lo1: flags=8008<LOOPBACK,MULTICAST> mtu 33224
fxp0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        address: 00:50:8b:01:cf:87
        media: Ethernet autoselect (100baseTX full-duplex)
        status: active
        inet 132.248.102.225 netmask 0xffffff00 broadcast 132.248.102.255
        inet6 fe80::250:8bff:fe01:cf87%fxp0 prefixlen 64 scopeid 0x1
pflog0: flags=0<> mtu 33224
pfsync0: flags=0<> mtu 2020
sl0: flags=c010<POINTOPOINT,LINK2,MULTICAST> mtu 296
sl1: flags=c010<POINTOPOINT,LINK2,MULTICAST> mtu 296
ppp0: flags=8010<POINTOPOINT,MULTICAST> mtu 1500
ppp1: flags=8010<POINTOPOINT,MULTICAST> mtu 1500
tun0: flags=10<POINTOPOINT> mtu 3000
tun1: flags=10<POINTOPOINT> mtu 3000
enc0: flags=0<> mtu 1536
bridge0: flags=0<> mtu 1500
bridge1: flags=0<> mtu 1500
vlan0: flags=0<> mtu 1500
        address: 00:00:00:00:00:00
vlan1: flags=0<> mtu 1500
        address: 00:00:00:00:00:00
gre0: flags=9010<POINTOPOINT,LINK0,MULTICAST> mtu 1450
gif0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1280
        physical address inet 132.248.102.225 --> 132.248.102.201
        inet6 fe80::250:8bff:fe01:cf87%gif0 -> prefixlen 64 scopeid 0x12
        inet6 3ffe:8070:1022:1::1 -> prefixlen 64
gif1: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
gif2: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
gif3: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
```

Autoconfigurar IPv6 en openBSD

Para ello se necesita modificar los siguientes archivos:

```
/etc/rc.conf
ipv6_enable="YES"

/etc/rc.local
route add -inet6 default 3ffe:8070:1022::1
route6d

/etc/sysctl.conf
net.inet6.ip6.forwarding=1     # 1=Permit forwarding (routing) of packets
net.inet6.ip6.accept_rtadv=1   # 1=Permit IPv6 autoconf (forwarding must be 0)
```

Sólo se agregan esas líneas a los archivos y reiniciamos para que tome los valores y se autoconfigure.

4.5 Utilización de herramientas y comprobación del stack IPv6

En esta sección se muestra el funcionamiento de la red IPv6 en Cuautitlán. A continuación se presentarán los resultados con algunas herramientas (comandos) que son muy utilizadas en IPv4 pero que ahora tendrán una ligera modificación para que puedan servir para IPv6, por mencionar algunos comandos tenemos los siguientes: ping6, traceroute6, ssh, telnet, netstat, entre otras.

Comprobación de conexión por IPv6 medio del túnel:

La comprobación más común es haciendo ping en este caso al extremo remoto del túnel

```
[root@ipv6 greg]# ping6 3ffe:8070:1022::1
PING 3ffe:8070:1022::1(3ffe:8070:1022::1) 56 data bytes
64 bytes from 3ffe:8070:1022::1: icmp_seq=1 ttl=64 time=16.9 ms
64 bytes from 3ffe:8070:1022::1: icmp_seq=2 ttl=64 time=38.0 ms
64 bytes from 3ffe:8070:1022::1: icmp_seq=3 ttl=64 time=54.2 ms
64 bytes from 3ffe:8070:1022::1: icmp_seq=4 ttl=64 time=35.8 ms
64 bytes from 3ffe:8070:1022::1: icmp_seq=5 ttl=64 time=29.0 ms
64 bytes from 3ffe:8070:1022::1: icmp_seq=6 ttl=64 time=55.4 ms
64 bytes from 3ffe:8070:1022::1: icmp_seq=7 ttl=64 time=126 ms

--- 3ffe:8070:1022::1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6065ms
rtt min/avg/max/mdev = 16.934/50.911/126.923/33.459 ms
```

también se hace una traza hacia el otro extremo del túnel y nos arroja el siguiente resultado:

```
[root@ipv6 greg]# traceroute6 3ffe:8070:1022::1
```

traceroute to 3ffe:8070:1022::1 (3ffe:8070:1022::1) from 3ffe:8070:1022::2, 30 hops max, 16 byte packets
*1 3ffe:8070:1022::1 (3ffe:8070:1022::1) 33.13 ms * 30.999 ms*

El cual es cierto ya que sólo hay un paso para llegar al otro extremo

Comprobación de la conexión por IPv6 hacia el mundo IPv6:

La comprobación del stack IPv6 se hará por medio de comandos y utilizando las páginas web con soporte IPv6. Así que primero se escoge un sitio IPv6 y se hace un ping al sitio IPv6 remoto:

[root@ipv6 greg]# ping6 www.6bone.net
PING www.6bone.net(www.6bone.net) 56 data bytes
64 bytes from www.6bone.net: icmp_seq=1 ttl=61 time=146 ms
64 bytes from www.6bone.net: icmp_seq=2 ttl=61 time=145 ms
64 bytes from www.6bone.net: icmp_seq=3 ttl=61 time=159 ms
64 bytes from www.6bone.net: icmp_seq=4 ttl=61 time=145 ms
64 bytes from www.6bone.net: icmp_seq=5 ttl=61 time=152 ms

--- www.6bone.net ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4039ms
rtt min/avg/max/mdev = 145.117/149.632/159.223/5.464 ms

Ahora se hace una traza hacia el mismo sitio IPv6

[root@ipv6 greg]# traceroute6 www.6bone.net
traceroute to 6bone.net (3ffe:b00:c18:1::10) from 3ffe:8070:1022::2, 30 hops max, 16 byte packets
*1 3ffe:8070:1022::1 (3ffe:8070:1022::1) 13.167 ms * 7.404 ms*
2 3ffe:1cff:0:f4::1 (3ffe:1cff:0:f4::1) 111.631 ms 110.451 ms 113.348 ms
3 rap.ipv6.viagenie.qc.ca (3ffe:b00:c18:1:290:27ff:fe17:fc0f) 144.457 ms 144.414 ms 144.096 ms
4 www.6bone.net (3ffe:b00:c18:1::10) 144.637 ms 153.451 ms 145.696 ms

Aquí se puede observar que son 4 saltos para llegar a nuestro destino, la dirección IPv6 y el tiempo en que llego al destino.

Se hace un ping y trazas hacia otros sitios IPv6:

[root@ipv6 greg]# ping6 -n www.kame.net
PING www.kame.net(2001:200:0:8002:203:47ff:fea5:3085) 56 data bytes
64 bytes from 2001:200:0:8002:203:47ff:fea5:3085: icmp_seq=1 ttl=54 time=415 ms
64 bytes from 2001:200:0:8002:203:47ff:fea5:3085: icmp_seq=2 ttl=55 time=387 ms
64 bytes from 2001:200:0:8002:203:47ff:fea5:3085: icmp_seq=3 ttl=55 time=388 ms

64 bytes from 2001:200:0:8002:203:47ff:fea5:3085: icmp_seq=4 ttl=55 time=429 ms
64 bytes from 2001:200:0:8002:203:47ff:fea5:3085: icmp_seq=5 ttl=55 time=516 ms

--- www.kame.net ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4036ms
rtt min/avg/max/mdev = 387.032/427.270/516.126/47.308 ms

[root@ipv6 greg]# ping6 -n www.ipv6.elmundo.es
PING www.ipv6.elmundo.es(2001:450:9:10::71) 56 data bytes
64 bytes from 2001:450:9:10::71: icmp_seq=1 ttl=58 time=414 ms
64 bytes from 2001:450:9:10::71: icmp_seq=2 ttl=58 time=464 ms
64 bytes from 2001:450:9:10::71: icmp_seq=3 ttl=58 time=418 ms
64 bytes from 2001:450:9:10::71: icmp_seq=4 ttl=58 time=469 ms

[root@ipv6 greg]# traceroute6 www.ipv6.elmundo.es
traceroute to www.ipv6.elmundo.es (2001:450:9:10::71) from 3ffe:8070:1022::2, 30 hops max, 16 byte packets
*1 3ffe:8070:1022::1 (3ffe:8070:1022::1) 7.504 ms * 7.619 ms*
2 3ffe:8070:1:13::2 (3ffe:8070:1:13::2) 63.007 ms 63.349 ms 62.88 ms
*3 3ffe:2900:f:e::2 (3ffe:2900:f:e::2) 180.283 ms 180.852 ms **
4 2001:450:1:1::21 (2001:450:1:1::21) 332.271 ms 333.541 ms 331.708 ms
5 2001:450:1:2001::a3 (2001:450:1:2001::a3) 361.76 ms 360.057 ms 360.796 ms
6 2001:450:9:10::71 (2001:450:9:10::71) 362.525 ms 361.077 ms 361.693 ms

[root@ipv6 greg]# traceroute6 www.kame.net
traceroute to orange.kame.net (2001:200:0:8002:203:47ff:fea5:3085) from 3ffe:8070:1022::2, 30 hops max, 16 byte packets
*1 3ffe:8070:1022::1 (3ffe:8070:1022::1) 6.666 ms * 6.957 ms*
2 3ffe:8070:1:13::2 (3ffe:8070:1:13::2) 68.427 ms 70.714 ms 126.436 ms
3 paix6.ttnet.ad.jp (3ffe:80a::e) 71.8 ms 71.555 ms 72.455 ms
4 2001:2a0:0:bb0a::1 (2001:2a0:0:bb0a::1) 174.962 ms 177.632 ms 175.637 ms
5 2001:2a0:0:bb04::6 (2001:2a0:0:bb04::6) 176.531 ms 177.302 ms 175.17 ms
6 hitachi1.otemachi.wide.ad.jp (2001:200:0:1800::9c4:2) 180.504 ms 180.365 ms 180.467 ms
7 pc3.yagami.wide.ad.jp (2001:200:0:1c04::1000:2000) 180.863 ms 180.578 ms 180.758 ms
8 gr2000.k2c.wide.ad.jp (2001:200:0:4819::2000:1) 181.72 ms 183.043 ms 183.056 ms
9 orange.kame.net (2001:200:0:8002:203:47ff:fea5:3085) 182.312 ms 182.208 ms 181.41 ms

Con eso se comprueba la salida hacia el mundo IPv6.

Comprobación de conexión por IPv6 para windowsXP

Comprobar la conexión a sitios que soportan IPv6 ejecutar el siguiente comando:

```
C:\>ping6 -n www.kame.net
PING www.kame.net(2001:200:0:8002:203:47ff:fea5:3085) 56 data bytes
64 bytes from 2001:200:0:8002:203:47ff:fea5:3085: icmp_seq=1 ttl=54 time=415 ms
64 bytes from 2001:200:0:8002:203:47ff:fea5:3085: icmp_seq=2 ttl=55 time=387 ms
64 bytes from 2001:200:0:8002:203:47ff:fea5:3085: icmp_seq=3 ttl=55 time=388 ms
64 bytes from 2001:200:0:8002:203:47ff:fea5:3085: icmp_seq=4 ttl=55 time=429 ms
64 bytes from 2001:200:0:8002:203:47ff:fea5:3085: icmp_seq=5 ttl=55 time=516 ms
```

Comprobación del soporte IPv6 en Internet Explorer de windowsXP

Para revisar si el navegador Internet Explorer ya soporta IPv6, abrir una página web que muestre en forma visual este soporte, por ejemplo:

http://www.ipv6forum.com
http://www.kame.net

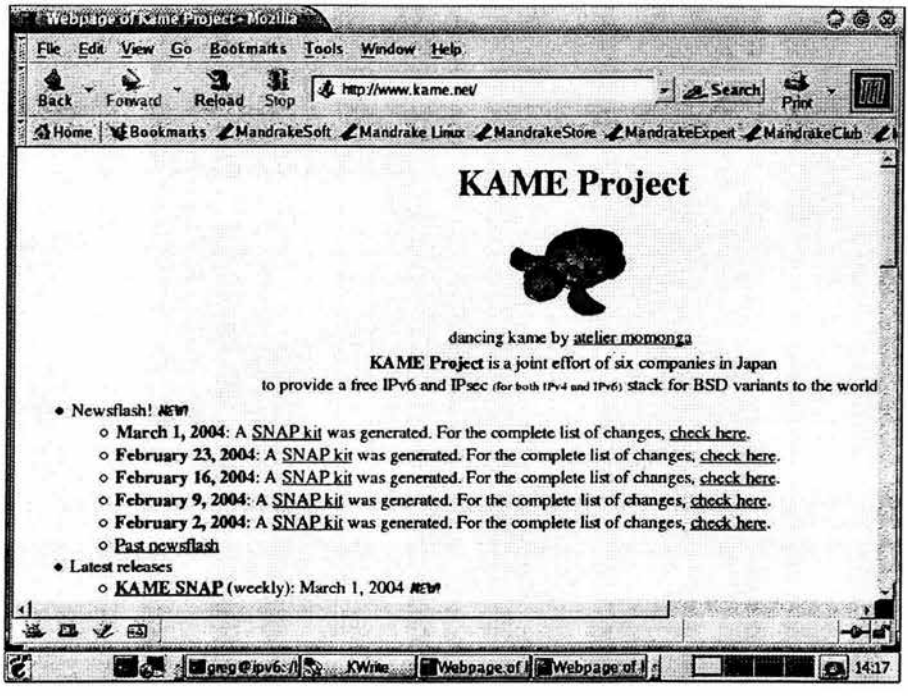


Figura 4.4 Proyecto Kame

El navegador **YA** soporta IPv6, se notará que gira el mundo de la imagen de la parte superior izquierda de la pantalla ó que la tortuga tiene movimiento. Este movimiento de las imágenes es solo representativo y una manera de distinguir consultas por IPv6.

Comprobación de conexión por IPv6 por medio de paginas web:

Usando el navegador que trae por *default* Linux que es Mozilla se consultaron algunas páginas con soporte IPv6 y se obtuvo lo siguiente:

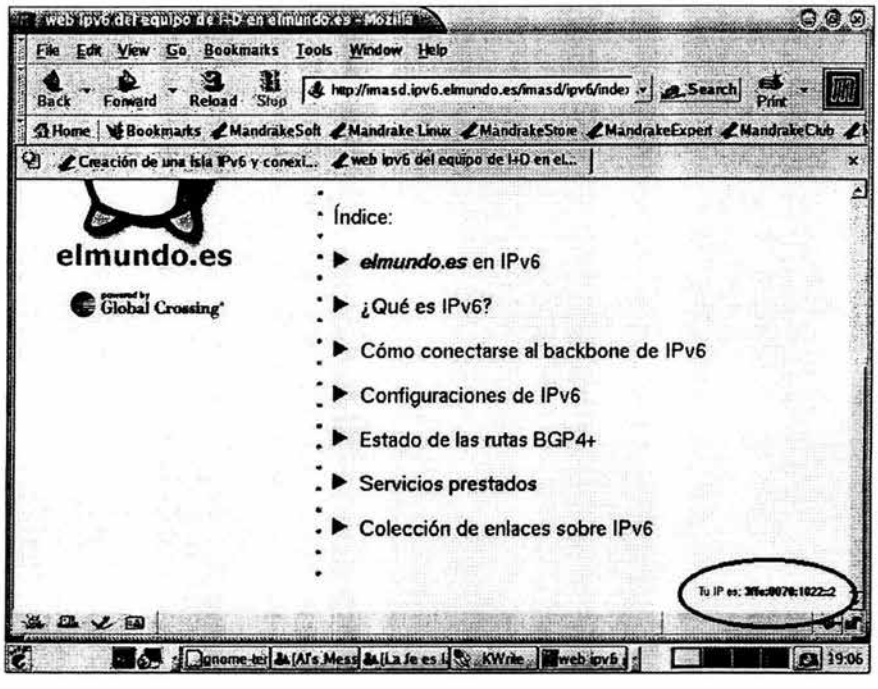


Figura 4.5 Navegación de pagina web "el mundo es" con soporte IPv6

Se puede notar que en la parte inferior derecha de la figura 4.5, se encuentra la dirección IPv6 que se configuró en la PC principal. Existen páginas que detectan automáticamente la dirección IP de la PC, ya sea la versión 4 o la versión 6.

Se entró a otra página que también tiene soporte IPv6, el resultado esta en la siguiente figura:

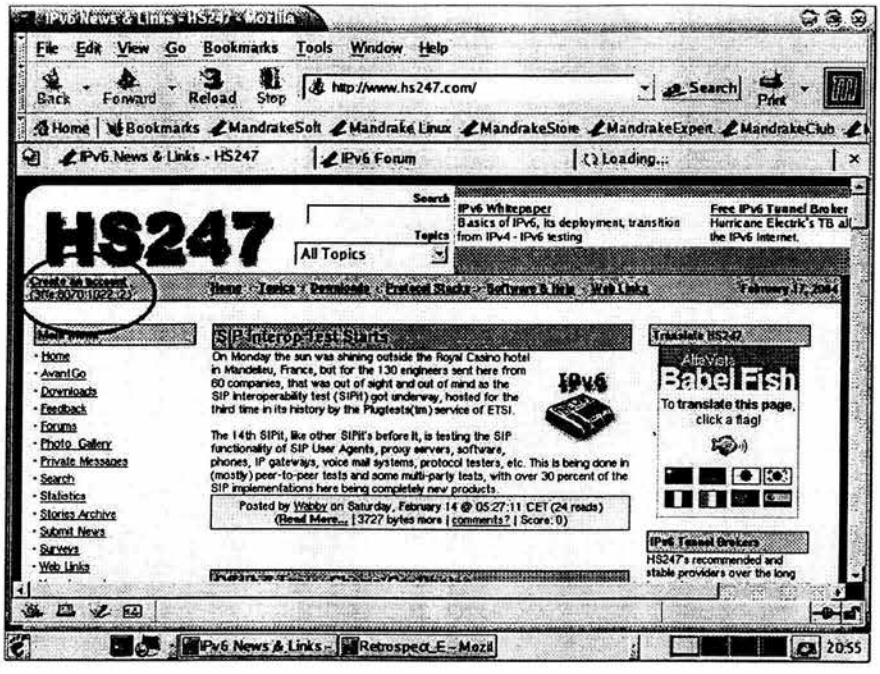


Figura 4.6 Navegación con IPv6

Igualmente que en el caso anterior, la página muestra la dirección IPv6 de mi PC.

En la página del proyecto Kame hay una tortuga que cuando uno entra con una dirección IPv6 la tortuga tiene movimiento, eso indica que estamos navegando con IPv6. Igualmente pasa cuando navegamos en la página de IPv6 Forum , cuando se navega con IPv6 el mundo que tiene representado, gira. Se tiene el caso de la página web de el "mundo es" que solo muestra la dirección IPv6 de nuestro host.

Herramientas para la comprobación de la conexión por IPv6

Algunas herramientas ya se probaron tal es el caso de ping y traceroute para la versión 6, sin embargo, se tienen más herramientas que a continuación presentaré.

Checando el DNS para la resolución de direcciones IPv6 con el siguiente comando nos muestra dicha resolución.

[root@ipv6 greg]# host -t AAAA www.join.uni-muenster.de
www.join.uni-muenster.de is an alias for tolot.join.uni-muenster.de.

tolot.join.uni-muenster.de has AAAA address
2001:638:500:101:2e0:81ff:fe24:37c6

Se ejecuta el comando telnet

[root@ipv6 greg]# telnet 3ffe:400:100::1 80
Trying 3ffe:400:100::1...
telnet: connect to address 3ffe:400:100::1: No route to host

Ahora se prueba con el Secure Shell (ssh), habilitado para IPv6:

[root@ipv6 greg]# ssh -6 ::1
root@::1's password:
Last login: Mon Feb 2 18:46:17 2004

con el siguiente comando se muestran las direcciones IPv4 e IPv6 así como las interfaces correspondientes:

[root@ipv6 greg]# ip addres show
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 brd 127.255.255.255 scope host lo
inet6 ::1/128 scope host
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 100
link/ether 00:01:80:2a:61:ca brd ff:ff:ff:ff:ff:ff
inet 132.248.102.201/24 brd 132.248.102.255 scope global eth0
inet6 fe80::201:80ff:fe2a:61ca/64 scope link
inet6 3ffe:8070:1022:1::1/64 scope global
3: sit0@NONE: <NOARP,UP> mtu 1480 qdisc noqueue
link/sit 0.0.0.0 brd 0.0.0.0
inet6 ::127.0.0.1/96 scope host
inet6 ::132.248.102.201/96 scope global
inet6 ::132.248.249.2/96 scope global
4: sit1@NONE: <POINTOPOINT,NOARP,UP> mtu 1480 qdisc noqueue
link/sit 0.0.0.0 peer 132.248.108.254
inet6 3ffe:8070:1022::2/64 scope global
inet6 fe80::84f8:66c9/64 scope link
inet6 fe80::84f8:f902/64 scope link

Para que nos muestre los túneles existentes:

[root@ipv6 IPv6]# ip -f inet6 tunnel show
sit0: ipv6/ip remote any local any ttl 64 nopmtudisc
sit1: ipv6/ip remote 132.248.108.254 local any ttl 64

Para ver la tabla de ruteo:

```
[root@ipv6 greg]# route -A inet6
Kernel IPv6 routing table
Destination                    Next Hop           Flags Metric Ref    Use Iface
::1/128                        ::                 U     0      6        0 lo
::127.0.0.1/128                ::                 U     0      0        0 lo
::132.248.102.201/128          ::                 U     0      0        0 lo
::132.248.249.2/128            ::                 U     0      0        0 lo
3ffe:8070:1022::2/128          ::                 U     0      31238    1 lo
3ffe:8070:1022::/64            ::                 UA    256    577      1 sit1
3ffe:8070:1022:1::1/128        ::                 U     0      0        0 lo
3ffe:8070:1022:1::/64          ::                 UA    256    0        0 eth0
3ffe::/16                      ::                 U     1      0        0 sit1
2000::/3                       3ffe:8070:1022::1  UG    1      65       0 sit1
2000::/3                       ::                 U     1      0        0 sit1
fe80::84f8:66c9/128            ::                 U     0      0        0 lo
fe80::84f8:f902/128            ::                 U     0      0        0 lo
fe80::201:80ff:fe2a:61ca/128   ::                 U     0      0        0 lo
fe80::204:75ff:fe90:2a87/128   ::                 U     0      0        0 lo
fe80::/64                      ::                 UA    256    0        0 eth0
fe80::/64                      ::                 UA    256    0        0 sit1
ff00::/8                       ::                 UA    256    0        0 eth0
ff00::/8                       ::                 UA    256    0        0 sit1
```

Para ver la salida de "route" para IPv6

```
[root@ipv6 IPv6]# ip -f inet6 route
unreachable ::/96 dev lo  metric 1024  error -101 mtu 16436 advmss 16376
unreachable 2002:c0a8::/32 dev lo  metric 1024  error -101 mtu 16436 advmss
16376
unreachable 2002:e000::/19 dev lo  metric 1024  error -101 mtu 16436 advmss
16376
3ffe:8070:1022::/64 via :: dev sit1  proto kernel  metric 256  mtu 1480 advmss
1420
3ffe:8070:1022:1::/64 dev eth0  proto kernel  metric 256  mtu 1500 advmss
1440
unreachable 3ffe:ffff::/32 dev lo  metric 1024  error -101 mtu 16436 advmss
16376
3ffe::/16 dev sit1  metric 1  mtu 1480 advmss 1420
2000::/3 via 3ffe:8070:1022::1 dev sit1  metric 1  mtu 1480 advmss 1420
2000::/3 dev sit1  metric 1  mtu 1480 advmss 1420
fe80::/64 dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440
fe80::/64 via :: dev sit1  proto kernel  metric 256  mtu 1480 advmss 1420
ff00::/8 dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440
ff00::/8 dev sit1  proto kernel  metric 256  mtu 1480 advmss 1420
unreachable default dev lo  metric -1  error -101
```

netstat es un comando muy socorrido por los administradores de red para ver el estado de las conexiones de red activas con características que son de gran

ayuda para la administración. Aquí se muestra este comando con dos opciones diferentes:

```
[root@ipv6 greg]# netstat -tan
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:32768         0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:677             0.0.0.0:*               LISTEN
tcp        0      0 :::443                  :::*                    LISTEN
tcp        0      0 3ffe:8070:1022::2:33648 2001:200:0:1003:207::80 ESTABLISHED
tcp        0      0 3ffe:8070:1022::2:33656 3ffe:400:280::1:80      ESTABLISHED
tcp        0      0 3ffe:8070:1022::2:33638 2001:618:1401::4:80     ESTABLISHED
tcp        0      0 3ffe:8070:1022::2:33648 2001:200:0:1003:207::80 ESTABLISHED
tcp        0      0 3ffe:8070:1022::2:33656 3ffe:400:280::1:80      ESTABLISHED
tcp        0      0 3ffe:8070:1022::2:33638 2001:618:1401::4:80     ESTABLISHED
```

con esta opción se ve el proceso y el nombre del servicio, además de las opciones que se vio en el caso anterior

```
[root@ipv6 greg]# netstat -lnptu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 127.0.0.1:32768         0.0.0.0:*               LISTEN      1512/xinetd
tcp        0      0 132.248.102.201:53      0.0.0.0:*               LISTEN      1445/
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN      1445/
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      1683/
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN      1445/
tcp        0      0 0.0.0.0:7741            0.0.0.0:*               LISTEN      1980/lisa
tcp        0      0 :::2601                 :::*                    LISTEN      2365/
tcp        0      0 :::2605                 :::*                    LISTEN      11948/
tcp        0      0 :::80                   :::*                    LISTEN      1872/httpd2
tcp        0      0 :::22                   :::*                    LISTEN      1487/sshd
tcp        0      0 :::443                  :::*                    LISTEN      1872/httpd2
udp        0      0 0.0.0.0:520             0.0.0.0:*                           1466/routed
udp        0      0 132.248.102.201:53      0.0.0.0:*                           1445/
udp        0      0 127.0.0.1:53            0.0.0.0:*                           1445/
```

Con el siguiente comando, muestra las interfaces con direcciones multicast :

```
[root@ipv6 greg]# ip -f inet6 maddr
1:      lo
        inet6 ff02::1
2:      eth0
        inet6 ff02::2
        inet6 ff02::1:ff00:1
        inet6 ff02::1:ff2a:61ca
```

```
      inet6 ff02::1
3:    eth1
      inet6 ff02::2
      inet6 ff02::1:ff00:1
      inet6 ff02::1:ff90:2a87
      inet6 ff02::1
4:    sit0
      inet6 ff02::1
5:    sit1
      inet6 ff02::1
```

Para listar los archivos abiertos así como su proceso, que puerto esta utilizando y varias opciones más. Este comando se usará para verificar que se encuentre funcionando efectivamente el "demonio" de bgp

```
root@ipv6 etc]# lsof -i
bgpd    11948  user   4u  IPv6 127581      TCP *:bgp (LISTEN)
bgpd    11948  user   7u  IPv6 127583      TCP *:nsc-posa (LISTEN)
bgpd    11948  user   8u  IPv6 127587      TCP [3ffe:8070:1022::2]:33391-
>[3ffe:8070:1022::1]:bgp (ESTABLISHED)
```

Ahora referente al ruteador IPv6, se configuró para que anuncie nuestra red, así que por medio de unos comandos se verán los resultados.

Para saber sobre nuestro anuncio, le damos el siguiente comando:

```
bgpd# sh ipv6 bgp 3ffe:8070:1022::1
BGP routing table entry for 3ffe:8070:1022::/48
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  3ffe:8070:1022::1
  Local
    :: from :: (132.248.102.201)
      Origin IGP, metric 0, localpref 100, weight 32768, valid, sourced, local, best
      Last update: Wed Feb 25 13:34:20 2004
```

Tecleando el siguiente comando se puede saber si efectivamente se encuentra la conexión con BGP esta establecida. La salida muestra la conexión de BGP de nuestros vecinos y cuantas redes se están recibiendo por ese vecino.

```
bgpd# sh ipv6 bgp summ
BGP router identifier 132.248.102.201, local AS number 278
376 BGP AS-PATH entries
0 BGP community entries

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
3ffe:8070:1022::1
            4   278    3454    1694        0    0    0 08:00:21      448
```

Total number of neighbors 1

Luego se checa que redes pertenecientes a la FES-C esta anunciando, el cual arroja el siguiente resultado:

```
bgpd# sh ipv6 bgp neighbors 3ffe:8070:1022::1 advertised-routes
BGP table version is 0, local router ID is 132.248.102.201
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 3ffe:8070:1022::/48
                    ::                       0    100  32768 i

Total number of prefixes 1
```

El cual es correcto ya que se le configuró para que anunciara todo el bloque que le fue asignado a la FES-Cuautitlán.

Para saber que redes se reciben o cuales está anunciando nuestro vecino, se teclea el siguiente comando:

```
bgpd# sh ipv6 bgp neighbors 3ffe:8070:1022::1 received-routes
BGP table version is 0, local router ID is 132.248.102.201
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 2001:200::/32    3ffe:8070:1:13::2
                                         100      0 6939 4716 2500 i
*> 2001:208::/32    2001:448:3:153::1
                                         100      0 18592 11537 7610 i
*> 2001:218::/32    2001:418:0:4000::1
                                    4    100      0 2914 i
*> 2001:220::/35    3ffe:8070:1:13::2
                                         100      0 6939 6939 3748 9270 i
*> 2001:228::/35    3ffe:8070:1:13::2
                                         100      0 6939 2516 2915 i
*> 2001:230::/35    3ffe:8070:1:13::2
*> 2001:251::/32    2001:448:3:153::1
                                         100      0 18592 11537 7660 9407 4538 23911
23912 i
*> 2001:258::/32    2001:418:0:4000::1
                                    4    100      0 2914 2510 i
*> 2001:260::/32    2001:418:0:4000::1
                                    4    100      0 2914 2518 ?
*> 2001:268::/32    3ffe:8070:1:13::2
                                         100      0 6939 2516 i
```

```
*> 2001:270::/32    3ffe:1cff:0:f4::1
                                100     0 237 10566 3786 i
*> 2001:288::/32    3ffe:8070:1:11::2
                                100     0 6342 109 17717 i
*> 2001:290::/32    3ffe:8070:1:13::2
                                100     0 6939 6939 3748 17832 17846 i
*> 2001:298::/32    3ffe:8070:1:13::2
                                100     0 6939 2516 9600 i
*> 2001:320::/32    3ffe:8070:1:13::2
                                100     0 6939 293 3425 17579 i
```

etcétera

Son 448 anuncios recibidos hasta este momento, hay que tener presente que cada día se unen más redes a IPv6.

Con esto se da por entendido que la implementación ha sido exitosa. Ahora sólo resta comenzar a levantar servicios que soporten el protocolo IPv6 y seguir haciendo pruebas, como por ejemplo, IPsec, servidor web, desarrollo de programas de red basados en IPv6, IPv6 móvil, Voz sobre IPv6, multicast, IPv6 sobre Internet2, Calidad de Servicio, entre muchas otras.

Conclusiones

Conclusiones

En el presente trabajo se habló de manera general de las características de IPv4 y sobre todo se trató con más detalle las características de IPv6 así como el procedimiento para la implementación en FES-Cuautitlán campo 4.

Las primeras pruebas no tuvieron éxito debido a que en un principio se trató de instalar el stack de IPv6 en una máquina de la marca SUN UltraSparc10, existe una versión en Linux que soporta esta plataforma, sin embargo, se pudo comprobar que no se encuentra lo suficientemente soportado el stack de IPv6 para esta plataforma. Con esto se demuestra que no todas las versiones ni todas las plataformas de sistemas operativos esta soportado el stack de IPv6 pero es un hecho que cada versión que van liberando procuran madurar el stack IPv6.

El procedimiento para dicha implementación se realizó siguiendo los pasos que el propio staff de IPv6 de la UNAM sugiere. Para realizar la implementación se utilizó una versión de Linux más reciente, en este caso se instaló Linux Mandrake 9.1 con una versión del kernel 2.4.21-0.13mdk y con ello se aseguró el buen funcionamiento del stack IPv6. De esta manera se procedió a la configuración del túnel hacia Cómputo Académico mejor conocido por sus siglas D.G.S.C.A.(Dirección General de Servicios de Cómputo Académico) teniendo la respuesta esperada.

Para tener una idea de lo que se quería instalar y cómo hacerlo se diseño un esquema para la implementación y otro esquema para el direccionamiento, para tener una base de donde partir.

En un inicio fue complejo el manejo de las direcciones IPv6 pero poco a poco uno se fue familiarizando con ésta nueva sintaxis de direccionamiento. Para realizar el esquema de direccionamiento, primero se tuvo que comprender la estructura y características de IPv6 y tener muy en cuenta que se maneja un sistema de numeración hexadecimal. Una vez que se tuvo claro esto, se empezó a direccionar creando una subred y a partir de ahí se configuraron las direcciones IPv6 a los clientes, utilizando los dos métodos, manual y automático.

Para el ruteo se utilizó "Quagga" que es la versión más nueva de Zebra. Uno de los protocolos utilizados es BGP4+ para que anuncie la red que se configuró para la FES-C, de esta manera, el ruteador conocerá otras redes IPv6 tanto de producción como de pruebas y finalmente tendrán salida los clientes al 6bone, también se realiza la autoconfiguración de los clientes.

Una característica más de IPv6 es la autoconfiguración, para ello se debe tener en cuenta que para hacer efectiva la autoconfiguración, se realizó la configuración llamada "stateless" dónde interviene el ruteador para asignar direcciones IPv6 a los clientes, los cuales con sólo habilitar el módulo de IPv6 se autoconfiguran.

Ahora bien, para no tener que estar configurando el túnel, las direcciones IPv6, la salida a Internet cada vez que se reinicie la computadora, se realizo una forma de configuración en el equipo para que cuando inicie la maquina lea ciertos archivos y cargue automáticamente toda la configuración necesaria para levantar el stack de IPv6, así como el túnel y los archivos relacionados con el ruteo.

Posteriormente, se realizaron pruebas de conexión utilizando ping6, traceroute6 hacia la UNAM y hacia otros sitios donde esta instalado IPv6. También se acceso a sitios web donde soportan el protocolo IPv6. Todas estas pruebas fueron exitosas

Con lo anterior se obtuvo experiencia en los sistemas operativos como es el caso de openBSD y Linux a nivel administración, también se obtuvo experiencia en la creación de túneles en los sistemas operativos mencionados incluyendo Windows. Cada uno tiene su propia forma de funcionamiento pero al final se logro lo esperado que fue la habilitación de IPv6 en dichos sistemas operativos.

Académicamente me aportó muchos conocimientos principalmente en la investigación y posteriormente en la práctica dónde se aplico toda la teoría que sirvió de base para la implementación de éste protocolo. Profesionalmente fue muy satisfactorio llevar esta propuesta a la realidad ya que como todo, siempre hay problemas que se tienen que resolver y que en ocasiones son complejas pero al fin y al cabo se encontraron soluciones óptimas.

Finalmente con éste trabajo de investigación y aplicación de ésta propuesta en la FES-C se obtuvo mayor experiencia y madurez sobre el tema, espero que se abra una puerta para futuras tecnologías, con ello probar y desarrollar nuevas aplicaciones relacionado con IPv6. Hace falta todavía mucho por hacer, esta implementación es sólo un paso, de aquí en adelante se pueden realizar muchas pruebas sobre este protocolo, se pueden implantar, por ejemplo, servicios de web, seguridad, H.323, VoIPv6, Multicast, IPv6 e Internet2 y probar las características intrínsecas que tiene éste protocolo y sobretodo que este trabajo sirva como pauta para futuros proyectos.

Glosario

Glosario de términos

- A -

Ancho de banda *Bandwidth* Margen de frecuencias capaz de transmitirse por una red de telecomunicación y de interpretarse en sus terminales.

Anycast. Cualquier difusión. Es un identificador para un conjunto de interfaces, típicamente pertenecen a diferentes nodos. Un paquete enviado a una dirección anycast es entregado a cualquiera de las interfaces identificadas con dicha dirección.

ATM *Asynchronous Transfer Mode*. Modo de transferencia asíncrono. Estándar internacional para conmutación de celdas, en el que se transportan varios tipos de servicios (voz, vídeo y datos) por medio de celdas de longitud fijas (53 bytes).

- B -

Backbone / Red básica. Red de transmisión (o "espina dorsal") a través de la cual se transportan datos de los diferentes nodos que están conectados a ella.

BGP *Border Gateway Protocol*. Protocolo de enrutamiento interdominios que reemplaza a EGP. BGP intercambia información de accesibilidad con otros sistemas BGP.

Bps. Abreviatura de bits por segundo, la medida estándar para las velocidades de transmisión de información. El número de bits de datos por señal multiplicado por los baudios, da como resultado el número de bits por segundo. Solamente en el caso de que cada estado de una línea esté representado por un bit, coincidirán la velocidad en baudios y en bits por segundo.

Bridge. Puente. Dispositivo que interconecta redes de área local (LAN) en la capa de enlace de datos OSI. Filtra y retransmite tramas según las direcciones a Nivel MAC.

Broadcast. Tipo de comunicación basada en la difusión en que todo posible receptor es alcanzado por una sola transmisión.

Bucle. Ruta donde los paquetes nunca alcanzan su destino, sino que pasan por ciclos repetidamente a través de una serie constante de nodos de red.

Buffer. Àrea de la memoria que se utiliza para almacenar datos temporalmente durante una sesión de trabajo.

Byte. Unidad de información utilizada por las computadoras. Cada byte está compuesto por ocho bits y representan un carácter.

- C -

Canal. Ruta de transmisión de comunicaciones a través de cualquier clase de medio de transmisión: cable conductor, radio, fibra óptica o de cualquier otro tipo.

CSMA/CD *Carrier Sensing Multple Access/Collision Detection*. Acceso múltiple con detección de portadora y detección de colisiones. Mecanismo de acceso al canal en el cual los dispositivos que desean transmitir primero verifican la existencia de portadora en el canal. Si no se detecta portadora en un cierto tiempo, los dispositivos pueden transmitir. Si dos dispositivos transmiten a la

vez, ocurre una colisión, que es detectada por los dispositivos en colisión, que retardan la retransmisión durante un período aleatorio. Este método es empleado por redes Ethernet y por IEEE 802.3

- D -

Datagrama. Usualmente se refiere a la estructura interna de un paquete de datos.

DNS *Domain Name Server.* Sistema de Nombres de Dominio. Es un sistema de base de datos distribuida que sirve para traducir nombres de computadoras a direcciones IP y viceversa.

- E -

Encabezado. Información de control colocada antes de los datos al encapsularlos para la transmisión en red.

Encapsulado *Encapsulation*. Es la función de empaquetado de datos en un encabezado particular de protocolos.

Enlace. Canal de comunicaciones de red que se compone de un circuito o ruta de transmisión y todo el equipo relacionado entre un emisor y un receptor. Se utiliza con mayor frecuencia para referirse a una conexión de WAN. A veces se denomina línea o enlace de transmisión.

Enrutamiento. Proceso de descubrimiento de una ruta hacia el host destino. El enrutamiento es sumamente complejo en grandes redes debido a la gran cantidad de destinos intermedios potenciales que debe atravesar un paquete antes de llegar a su destino

Ethernet. Red de área local de alta velocidad desarrollada en forma conjunta por Xerox, Intel y Digital Equipment Corporation que utiliza el protocolo CSMA/CD. Se ha convertido en un estándar de red corporativa.

- F -

FastEthernet. Cualquiera de varias especificaciones de Ethernet de 100-Mbps. FastEthernet ofrece un incremento de velocidad diez veces mayor que el de la especificación de Ethernet 10BaseT, aunque preserva características tales como formato de trama, mecanismos MAC, y MTU. Se basa en una extensión de la especificación IEEE 802.3. Ver también Ethernet.

Frame El término frame es utilizado para identificar un paquete de datos dentro de la capa de enlace de datos del modelo OSI. Los términos paquete, datagrama, segmento y mensaje también se emplean para describir agrupamientos lógicos de información en varias capas del modelo OSI.

Fragmentación. Proceso de dividir un paquete en unidades más pequeñas al transmitir a través de un medio de red que no puede acomodar el tamaño original del paquete.

FTP *file Transfer protocol.* Protocolo de transferencia de archivos. Protocolo que permite a un usuario de un sistema acceder y transferir a y desde otro sistema de una red

- G -

Gateway ***Pasarela***. También se conoce al término como Pasarela de Enlace. Una pasarela es un programa o dispositivo de comunicaciones que transfiere datos entre redes que tienen funciones

similares pero implantaciones diferentes. Hoy se utiliza el término router (direccionador, encaminador, enrutador) en lugar de la definición original de gateway. Dispositivo empleado para conectar redes que usan diferentes protocolos de comunicación de forma que la información puede pasar de una a otra..

- H -

H.323 Estándar de la ITU-T para voz y videoconferencia interactiva en tiempo real en redes de área local, LAN, e Internet

Hardware. A los componentes que es posible ver y tocar se les llama "hardware", palabra inglesa cuyo significado es máquina o "cosa dura".

Header Cabecera. Primera parte de un paquete de datos que contiene información sobre las características de este.

Hipertexto. Son documentos que contienen enlaces con otros documentos; al seleccionar un enlace automáticamente se despliega el segundo documento.

Host. Anfitrión. Sistema de cómputo en una red. Es similar a los términos device (dispositivo) o node (nodo), excepto que usualmente implica un sistema de cómputo, mientras que dispositivo y nodo generalmente se aplican a cualquier sistema en red, que incluye terminal servers (servidores de terminales) y enrutadores.

Hub. En general, dispositivo que sirve como centro de una topología en estrella. También denominado repetidor multipuerto. Dispositivo de hardware o software que contiene múltiples módulos de red y equipos de red independientes pero conectados. Los hubs pueden ser activos (cuando repiten señales que se envían a través de ellos) o pasivos (cuando no repiten, sino que simplemente dividen las señales enviadas a través de ellos).

HTTP *HyperText Transport Protocol.* (Protocolo de transporte de hipertexto). Protocolo para transferir archivos o documentos hipertexto a través de la red. Se basa en una arquitectura cliente/servidor.

- I -

IANA *Internet Assigned Number Authority.* Agencia de asignación de números de Internet. Antiguo registro central de diversos parámetros de los protocolos de Internet, tales como puertos, números de protocolo y empresa, opciones códigos y tipos. Fue sustituido en 1998 por ICANN

ICANN *Internet Corporation for Assigned Names and Numbers.* Corporación de Internet para la asignación de nombres y números. Organismo independiente sin ánimo de lucro con el objeto de gobernar, entre otras cosas, la asignación de espacio de direcciones IP y la gestión del sistema de asignación de nombres de dominio.

ICMP *Internet Control Messages Protocol.* Protocolo de Internet de control de mensajes. Proceso TCP/IP que proporciona el conjunto de funciones utilizado para el manejo y control de lasa capas de red.

IEEE *Institute of Electrical and Electronic Engineers.* Instituto de ingenieros eléctricos y electrónicos. Asociación de ingenieros que definen normas para estándares de comunicación.

IETF *Internet Engineering Task Force.* Fuerza de trabajo de ingeniería de Internet. Organismo encargado de proponer y establecer los estándares de Internet.

Interfaz. Interface. Es un punto de una vía de comunicación que permite el intercambio de información entre dos dispositivos o sistemas y para el que se han especificado sus características físicas, eléctricas y el tipo de señales a intercambiar, así como su significado.

Internet. Red de redes de cobertura mundial que están interconectadas entre sí a la cual están conectados millones de usuarios (personas, organismos y empresas) en todo el mundo. Internet provee servicios de transferencia de archivos, correo electrónico, sesiones remotas, noticias, entre otros.

IP *Internet Protocol.* Protocolo de Internet. Conjunto de reglas que regulan la transmisión de paquetes de datos a través de Internet. La versión actual es IPv4 y se intenta implementar la versión 6 (IPv6) que permitirá mejores servicios.

IPSec *IP Security,* Protocolo de Seguridad IP. Es un protocolo diseñado para proporcionar un nivel de seguridad en la capa IP.

IPX *Internet Packet Exchange.* Intercambio de paquetes en Internet. Un protocolo de comunicaciones del NetWare de Novell que se utiliza para encaminar mensajes de un nodo a otro

ISO *International Standards Organization.* Organización internacional para la normalización. Organización de carácter voluntario que es responsable de la creación de estándares internacionales en muchas áreas, incluyendo la informática y las comunicaciones.

ISP *Internet Service Provider.* Proveedor de Servicios Internet. Empresa encargada de ofrecer la infraestructura de acceso para que los clientes puedan conectar a Internet utilizando los medios de acceso estándar (módem, RTC, RDSI y ADSL).

ITU-T *International Telecommunications Union – Telecommunications.* Unión Internacional de Telecomunicaciones- Telecomunicaciones. Organismo internacional que desarrolla estándares para las diferentes tecnologías de telecomunicaciones a nivel mundial. La ITU-T realiza funciones que desempeñaba la CCITT.

- K -

Kilobyte Son mil bytes. Comúnmente ahora son 1024 (2^{10}) bytes.

Kbps Kilobits por segundo, unidad de transferencia de datos sobre un enlace.

- L -

LAN *Local Area network* (Red de área local). Conjunto de computadoras y otros dispositivos comunicados entre sí dentro de un área relativamente pequeña.

- M -

MAC *Media Access Control.* Control de Acceso a Medio. Protocolo que define las condiciones en las cuales las estaciones de trabajo acceden al medio. Su uso está difundido en las LAN. En las LAN tipo IEEE la capa MAC es la subcapa más baja del protocolo de la capa de enlace de datos.

Mbps Megabits por segundo: unidad de medida de la capacidad de transmisión por una línea de telecomunicación.

Megabyte Un millón de bytes.

MTU *Maximum transmission Unit.* Unidad de transmisión máxima. Se refiere al paquete de tamaño máximo, en bytes, que una interfaz en particular puede manejar.

Multidifusión *Multicast.* Es un identificador para un grupo de nodos. Un nodo puede pertenecer a uno o varios grupos multicast.

- N -

Networking. Interconexión de estaciones de trabajo, dispositivos periféricos (por ejemplo, impresoras, unidades de disco duro, escáneres y CD-ROM) y otros dispositivos.

NLA *Next Level Aggregation.* Agregación de Siguiente Nivel. En IPv6 se utiliza para identificar un cliente en especifico. Permite a un ISP crear varios niveles de jerarquía de direccionamiento dentro de una red para organizar el enrutamiento y el direccionamiento en un nivel inferior.

NOC *Network Operation Center.* Centro de Operaciones de la Red. Es un grupo de personas responsable de la operación diaria de la red. Cada proveedor de servicios tiene su propio NOC.

Nodo. Punto final de la conexión de red o una unión que es común para dos o más líneas de una red. Los nodos pueden ser procesadores, controladores o estaciones de trabajo. Los nodos, que varían en cuanto al enrutamiento y a otras aptitudes funcionales pueden estar interconectados mediante enlaces y sirven como puntos de control en la red. La palabra nodo a veces se utiliza de forma genérica para hacer referencia a cualquier entidad que tenga acceso a una red y frecuentemente se utiliza de modo indistinto con la palabra dispositivo.

- O -

OSPF *Open shortest Path First.* Primera trayectoria abierta más corta. Algoritmo de enrutamiento jerárquico IGP de estado de enlace

OSI *Open System Interconnection.* Interconexión abierta de sistemas. Programa internacional de estandarización apyado por ISO y la ITU-T para desarrollar estándares para redes de datos que facilita la interoperabilidad de equipos hechos por diversos fabricantes.

OUI *Organizational Unique Identifier.* Identificador organizativo único. Tres octetos que asigna el IEEE en un bloque de direcciones de 48 bits.

- P -

Paquete. Cantidad determinada de caracteres (octetos) que se toma como unidad, y dotada de una estructura definida de trama y de campos.

Password contraseña. Palabra o clave privada utilizada para confirmar una identidad en un sistema remoto que se utiliza para que una persona no pueda usurpar la identidad de otra.

PC *Personal Computer.* Computadora personal

PING *Packet Internet Groper.* Rastreador de Paquetes Internet. Programa utilizado para comprobar si un Host está disponible. Envía paquetes de control para comprobar si el anfitrión está activo y los devuelve.

Protocolo. Conjunto de reglas que gobiernan las comunicaciones entre sistemas de telecomunicación.

- Q -

QoS *Quality of Service.* Calidad de Servicio. Medida de desempeño de un sistema de transmisión que considera la calidad de la transmisión y la disponibilidad del servicio

- R -

RAM. Random Access Memory. Memoria de Acceso Aleatorio. Se les llama así porque es posible dirigirse directamente a la célula donde se encuentra almacenada la información. Su principal característica es que la información se almacena en ellas provisoriamente, pudiendo ser grabadas una y otra vez.

RFC *Request for Comments.* Solicitud para Comentarios. Es un conjunto de documentos en los cuales los estándares de Internet, los estándares propuestos y, generalmente, las ideas en proceso de aceptación son documentados y publicados.

RIP *Routing Information Protocol.* Protocolo de información de enrutamiento. IGP proporcionado con los sistemas UNIX BSD. RIP utiliza el número de saltos como métrica de enrutamiento. RIPng (RIP*next generation*) de siguiente generación es para la versión 6 del protocolo IP-

Router encaminador, enrutador, ruteador. Dispositivo que distribuye tráfico entre redes. La decisión sobre a dónde enviar los datos se realiza en base a información de nivel de red y tablas de direccionamiento. Es el nodo básico de una red IP.

- S -

Segmentación. Proceso de división de un solo dominio de colisión en dos o más dominios de colisión para reducir las colisiones y la congestión de la red.

Servidor *Server.* Sistema que trata las peticiones de datos, el correo electrónico, la transferencia de ficheros y otros servicios de red realizados por otros sistemas u ordenadores (clientes).

SMTP *Simple Mail Transfer Protocol.* Protocolo de transferencia simple de correo. Estándar para el intercambio de correo electrónico en internet que permite la interconexión de redes diferentes entre sí. Este protocolo es uno de los englobados en TCP/IP.

SNA *System Network architecture.* Arquitectura de red de sistema. Arquitectura de red desarrollada por IBM.

SNMP *Simple Network Management Protocol* (Protocolo simple de direccionamiento de red). Uno de los protocolos pertenecientes a la familia TCP/IP utilizado para gestionar grandes redes, internet por excelencia. SNMP se encarga de realizar funciones de direccionamiento de red.

Sniffer. Literalmente "Husmeador". Pequeño programa que busca una cadena numérica o de caracteres en los paquetes que atraviesan un nodo con objeto de conseguir alguna información. Normalmente su uso es ilegal.

Software. Esta palabra inglesa que significa "cosa suave", tiene dos significados: (a) uno amplio, de "procedimientos lógicos, para la cooperación armónica de un grupo de personas y máquinas,

persiguiendo un objetivo común"; (b) el otro restringido, de "programas de computadora", o conjunto de instrucciones, que se pone en la memoria de una computadora para dirigir sus operaciones.

SLA. *Site Level Aggregation*. Agregador de nivel de sitio. En IPv6 es usado por organizaciones finales para crear su propia estructura jerárquica de direcciones e identificar sus subredes.

Subred. Red segmentada en una serie de redes más pequeñas. En redes IP, una red que comparte una dirección de subred individual. Las subredes son redes segmentadas de forma arbitraria por el administrador de la red para suministrar una estructura de enrutamiento jerárquica, de varios niveles mientras protege a la subred de la complejidad de direccionamiento de las redes conectadas. A veces se denomina subnetwork.

Switch. Dispositivo que conecta computadoras. El switch actúa de manera inteligente. Puede agregar ancho de banda, acelerar el tráfico de paquetes y reducir el tiempo de espera. Para aislar la transmisión de una computadora a otra, los switches establecen una conexión temporal entre la fuente y el destino, y la conexión termina una vez que la conversación se termina.

- T -

Telecomunicaciones. Toda emisión, transmisión o recepción de signos, señales, escritos, imágenes, voz sonidos o información de cualquier naturaleza que se efectúa a través de hilos, radioelectricidad, medios ópticos, físicos, u otros sistemas electromagnéticos.

Telnet Protocolo/aplicación que permiten establecer una sesión remota con otras computadoras en Internet.

TLA *Top Level Aggregation*. Agregador de nivel superior. Se trata del nivel superior en la estructura jerárquica de enrutado de IPv6. Los TLA se asignan a grandes proveedores de internet. Se han asignado dos tipos: pTLA (*pseudo Top Level Aggregation)* para pruebas e investigación y sTLA (*sub Top Level Aggregation*) para producción.

Tráfico. Toda emisión, transmisión o recepción de signos, señales, datos, escritos, imágenes, voz, sonidos o información de cualquier naturaleza que se efectúe a través de una red de telecomunicaciones

TCP/IP Transmission Control Protocol/Internet Protocol Familia de protocolos que hacen posible la interconexión y tráfico de red en Internet. A ella pertenecen por ejemplo: FTP, SMTP, NNTP, etc.. Los dos protocolos más importantes son los que dan nombre a la familia IP y TCP.

- U -

UDP *User Datagram Protocol*. Protocolo de Datagrama de Usuario. Protocolo abierto en el que el usuario define su propio tipo de paquete.

Unicast. Se refiere a Protocolos o Dispositivos que transmiten los paquetes de datos de una dirección IP a otra dirección IP, dicho de otra manera, es un identificador para una sola interfaz.

UTP *Unshielded Twisted-Pair* Par trenzado no blindado. Medio de cable de cuatro pares que se emplea en varias redes. UTP no requiere el espacio fijo entre conexiones que es necesario para las conexiones de tipo coaxial. Hay cinco tipos de cableado UTP de uso común: cableado de Categoría 1, cableado de Categoría 2, cableado de Categoría 3, cableado de Categoría 4 y cableado de Categoría 5.

- V -

VoIP *Voice over IP* Voz sobre Protocolo de Internet (IP). La habilidad para transportar voz telefónica normal sobre una red de datos basada en el protocolo de Internet, con la misma funcionalidad, confiabilidad y calidad de voz que ofrecen las empresas telefónicas tradicionales.

- W -

WAN *Wide Area Network*. Red de Área Amplia. Normalmente expresada de forma abreviada y en inglés como "WAN", es una red de comunicaciones, de concepto análogo a LAN, pero en distancias mayores y por lo general con recurso a las redes públicas de telecomunicaciones para los enlaces entre distintas sedes.

WWW *World Wide Web*. Red de Alcance Mundial. Es un sistema basado en hipertexto cuya función es buscar y tener acceso a recursos de Internet.

Bibliografía

Bibliografía

McLean Ian
La Biblia de TCP/IP
España, Ed. Anaya Multimedia, 2001.

Forouzan A. Behrouz
TCP/IP Protocol Suite
Ed. Mc Graw-Hill, 2000.

Teare Diane
Designing Cisco Networks
Ed. Cisco Systems, 1999.

Gai S.
Internetworking IPv6 with Cisco Routers
Ed. Mc Graw-Hill, 1998.

Miller M.
Implementing IPv6, 2nd Edition
Estados Unidos, Ed. M&T Books, 2000.

Ammann Paul T.
Managing Dynamic IP Networks
Ed. Mc Graw Hill, 2000.

Maufer Thomas A.
IP Fundamentals
Upper Saddle River NJ, Ed. Prentice-Hall, 1999.

López Ángel, Novo Alejandro
Protocolos de Internet, Diseño e Implementación en Sistemas UNIX
Colombia, Ed. Alfaomega ra-ma,2000

Palmer Michael J.
Redes de Computadoras- Una guía práctica
México, Ed. Thompson Learning, 2001

Maxwell Steve
Red Hat Linux, Herramientas para la Administración de Redes
Colombia, Ed. Mc Graw Hill , 2001

Guía del primer año, Programa de estudios autorizados por la Academia de Networking de Cisco Systems, 2ª Edición
Cisco Press

Guía del segundo año, Programa de estudios autorizados por la Academia de Networking de Cisco Systems, 2ª Edición
Cisco Press

Microsoft
Fundamentos de Redes
2ª. Edición, Microsoft Press, 1998.

Ford, Merilee
Tecnologías de Interconectividad de Redes
Mexico, Prentice Hall, 1998.

Krol Ed
Conéctate al Mundo de Internet, Guía y Catálogo
México, McGraw-Hill, 1995.

Naik Dilip C.
Internet Standards and Protocols
USA, Microsoft Press, 1998.

Hagen Silvia
IPv6 Essentials
USA, O´Reilly & Associates, 2002

Albitz Paul, Liu Cricket
DNS AND BIND
3ª. Edición, Beijin, O´reilly, 1998

Cox Philip, Sheldon Tom
WINDOWS 2000, Manual de seguridad
México, McGraw-Hill, 2002.

Referencias

Página de la IETF
http://www.ietf.org/html.charters/ipngwg-charter.html

Página de la Red Mundial de IPv6
http://www.6bone.net/

Página del proyecto IPv6 en la UNAM
http://www.ipv6.unam.mx/

Página del Foro de IPv6
http://www.ipv6forum.org/

Página de la IEEE
http://www.ieee.org/rfcs.html

Proyecto Euro6IX
http://www.euro6ix.org

Páginas de Microsoft IPv6
http://msdn.microsoft.com/downloads/sdks/platform/tcpipv6.asp
http://msdn.microsoft.com/downloads/sdks/platform/tpipv6/faq.asp

Página del proyecto IPv6 Linux HOW-TO por Peter Bieringer
http://www.tldp.org/HOWTO/Linux+IPv6-HOWTO/

Página de Quagga
http://www.quagga.net
http://www.eduangi.com/quagga/docs.html

Página de El mundo es
http://imasd.elmundo.es/imasd/ipv6/cfg/router-freebsd.html

Página del proyecto KAME
http://www.kame.net

Página de HS247
http://www.hs247.com

Martínez Palet Jordi
Tutorial de IPv6
http://www.consulintel.es/Html/ForoIPv6/Documentos/ Tutorial%20de%20IPv6.pdf

Página del Grupo de trabajo de IPv6
Tutorial de IPv6.
http://www.ipv6.unam.mx/documentos/Tutorial-IPv6.pdf

Página de ARIN
http://www.arin.net/regserv/ipv6/IPv6.txt.

Página de RIPE
http://www.ripe.net/ripencc/about/regional/maps/ipv6policy-draft-090699.html

Página de APNIC
http://www.apnic.net/drafts/ipv6/ipv6-policy-280599.html