UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

IMPLEMENTACIÓN DE SERVICIOS PARA OFRECER CONECTIVIDAD IPv6 EN RedUNAM

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

PRESENTA:

JOSÉ GUADALUPE SERRATO CERVANTES

DIRECTOR DE TESIS:

ING. AZAEL FERNÁNDEZ ALCÁNTARA

MÉXICO, D.F, **2009**

AGRADECIMIENTOS FORMALES

Al proyecto *IPv6 de la UNAM* del cual pude obtener el apoyo para realizar la presente tesis.

Al Ing. Azael Fernández Alcántara por el apoyo, tiempo y guia durante la realización de este trabajo de investigación.

A la Universidad Nacional Autónoma de México y a la Facultad de Ingeniería por darme una formación en sus aulas e instalaciones, donde obtuve conocimiento, experiencias de profesores y compañeros que han hecho la persona que soy.

A la Dirección General de Servicios de Cómputo Académico, al personal academico que labora en ésta que me permitieron realizar esta tesis con los recuersos tecnicos necesarios, permitiendome desarrollar habilidades prácticas para ser un mejor profesionista.

A mis sinodales, M.C. María Jaquelina López Barrientos, M.I. Aurelio Adolfo Millán Nájera, Dra. Ana María Vázquez Vargas, M.I. Ángel César Govantes Saldívar y mi director de tesis Ing. Azael Fernández Alcantára, por sus observaciones y correcciones realizadas sobre este trabajo.

AGRADECIMIENTOS PERSONALES

A Dios.

A mis padres, María de la Paz Cervantes y Juan Carmen Serrato, a los cuales les debo todo lo que soy, que con su paciencia, amor, cariño y comprensión e podido realizar mis estudios en una de las mejores Universidades del país, a la cual queria pertener. Gracias!!!

A mis hermanos, Juan, Lourdes, Mariana y Monica que con su apoyo incondicional me han impulsado para concluir esta tesis.

A Nancy Gabriela Rodríguez, por tu apoyo, confianza, cariño, comprensión, amor que me has brindado durante el tiempo en el que realizace este trabajo y el que hemos compartido juntos. Gracias por estar en mi vida.

A mis compañeros de la Facultad de Ingeniería, Ing. Pedro Becerril, Ing. Gustavo Goméz, Ing. Alfredo Gutiérrez, Ing. Diana Sánchez, Arturo Hernández, Lucelde Fernández, Uriel Canto, Lidia Rivera, que en ellos encontre grandes amigos y comparti muchos momentos divertidos y de aprendizaje.

A mis amigos, Arturo, Daniel, Itzel, Laura, Alma Alejandra, Silvia Karina, Flor Beatriz, Nancy Claudia, Eden, Maricela y Liliana, han dejado una marca en mi vida, que de una manera directa o indirecta influyeron en mi formación y en la realización de este trabajo.

A los compañeros que compartiron conmigo este proceso en el NETLab, Luis Enrique, Nayelli, Marco, Giovanni e Israel, que con sus comentarios y apoyo colaboraron para el desarrollo de este trabajo.

Y por último a las personas las cuales no pude mencionar su nombre en estas cortas lineas, gracias a ustedes he podido concluir una etapa importante de mi vida profesional y sin su ayuda no hubiera podidos relizar varios logros en mi vida, gracias.

Índice General

Pag.

Índice de Figuras

Pag.

Capítulo 1

Capítulo 2

Capítulo 3

Capítulo 4

Capítulo 5

Capítulo 6

Capítulo 7

Capítulo 8

Anexo 2

Anexo 3

Índice de tablas

Capítulo 2

Capítulo 4

Capítulo 5

Capítulo 7

Capítulo 8

Anexo 1

Anexo 2

Anexo 3

INTRODUCCIÓN

En la actualidad las redes que conforman el Internet van en crecimiento día con día, es un hecho que ha aumentado el número de direcciones IPv4 en uso por lo que varios organismos han dado avisos de alerta sobre la inminente escasez de las mismas y han estimado fechas límite para su agotamiento. Aunque se han desarrollado métodos para extender su vida útil, el número de direcciones sigue siendo finito, y cada vez la disponibilidad es menor.

Es por esto que se diseñó una nueva versión del protocolo de Internet denominada IPv6, que es una actualización necesaria para solucionar el problema del número de direcciones disponibles y algunos otros problemas que se han venido presentando en IPv4, ya que el avance tecnológico, el crecimiento acelerado de las redes, la necesidad de mayor seguridad y un mejor protocolo que brinde robustez a las mismas, fueron los principales impulsores de esta nueva versión.

Dado el panorama actual, es bueno dar a conocer esta nueva versión del protocolo principalmente a los usuarios con conocimientos técnicos en redes ya que les es prácticamente desconocido, en cambio para la mayoría de los usuarios no técnicos de Internet tiene que ser transparente.

El objetivo de esta tesis es documentar e implementar servicios con soporte IPv6 para ofrecer conectividad nativa o por túneles a la comunidad universitaria de la RedUNAM por medio de algunos mecanismos de transición de IPv4 a IPv6, así como proponer actividades de difusión que ayuden a la transición a esta versión del protocolo de Internet.

También es necesario difundir, habilitar e implementar IPv6 en diferentes sistemas operativos cómo por ejemplo Windows Vista, Windows XP, FreeBSD, Debian y Ubuntu. Para construir escenarios de prueba que sirvan en la implementación de aplicaciones con soporte IPv6 y diversos servicios que ayuden a la transición a la nueva versión del protocolo. La elección de las plataformas anteriores se basó en que éstas son comúnmente utilizadas en los laboratorios de la Dirección General de Servicios de Cómputo Académico (DGSCA).

Esta tesis se organizó de tal manera que se den a conocer los conceptos sobre IPv6 de la forma más clara y precisa posible, por lo que en el capítulo 1 se exponen conceptos del modelo de referencia OSI y de la pila TCP/IP, con el fin de explicar a detalle el fundamento de la versión 4 del protocolo de Internet y cómo se estructura la interconexión entre computadoras.

En el capítulo 2 se describen los conceptos básicos sobre IPv6, se comentan cada uno de los campos que forman el encabezado de las direcciones, se explica el tipo de direcciones que existen en IPv6 así como su arquitectura, los diferentes mecanismos de transición de IPv4 a IPv6 que más se utilizan y las formas para obtener direcciones IPv6 en un segmento de red.

En el capítulo 3 se presentan los conceptos sobre los diferentes túneles que pueden utilizarse para la realización de conexiones entre equipos mediante IPv6 y la arquitectura de las direcciones para cada túnel.

En el capítulo 4 se describen los protocolos y los tipos de ruteo que actualmente se utilizan en pequeñas y grandes redes, su funcionamiento para la versión 4, los que cuentan con soporte para la versión 6 y una comparación de los mismos.

En el capítulo 5 se expone el funcionamiento de DNS con IPv6, los conceptos básicos que son requeridos para entender su operación y las diferentes herramientas para la manipulación de las direcciones IPv6.

En el capítulo 6 se presenta la evolución de IPv6 en la UNAM, los servicios que se ofrecen a la comunidad universitaria, las actividades actuales y los pendientes que es necesario realizar para difundir aún más el conocimiento y masificar la utilización de IPv6.

En el capítulo 7 se exponen los pasos que deben llevarse a cabo en toda planeación de una transición IPv4 a IPv6 particularizando el caso de RedUNAM, como los aspectos a considerar, los escenarios posibles, la realización de un inventario y la situación de servicios y aplicaciones en uso con IPv4, así como la selección del mecanismo de transición más adecuado y las pruebas previas recomendadas para poder implementar los servicios con soporte IPv6. Finalmente se expone la necesidad de implementar procedimientos y políticas para poder ofrecer direcciones IPv6.

En el capítulo 8 se describe la instalación y la configuración de los servicios implementados con soporte IPv6 mediante distintos métodos de transición como son túnel broker, Servidor teredo y un relay 6to4, para poder proporcionar conectividad IPv6 a los usuarios de RedUNAM.

En el anexo 1 se presenta la forma en la cual se debe de habilitar y configurar el soporte IPv6 en algunas plataformas para utilizar diversos mecanismos de transición. Así como la configuración de túneles manuales, túneles 6to4 y túneles teredo para obtener conectividad IPv6 y diversas pruebas de conectividad con el Internet IPv6.

En el anexo 2 se describe el uso del servidor túnel broker implementado en el NETLab (Laboratorio de Tecnologías Emergentes de Red) para poder obtener conectividad IPv6 de forma automática en RedUNAM, mediante este servicio.

En el anexo 3 se exponen los scripts de configuración automática programados para ser utilizados por los usuarios de RedUNAM, con los cuales se facilita el uso de los tres servicios que se implementaron en el laboratorio para ofrecer conectividad IPv6.

Finalmente se presentan las conclusiones del trabajo realizado a lo largo de los capítulos descritos, en donde se resumen las pruebas hechas y los servicios implementados para ser utilizados por los usuarios de RedUNAM y se sientan bases para poder continuar con diversas investigaciones que permitan adoptar más fácilmente la versión más reciente del protocolo de Internet IPv6.

Capítulo 1
Modelo de referencia OSI y pila TCP/IP

1.1 Modelo OSI.

El modelo para la interconexión de sistemas abiertos, OSI por sus siglas en inglés (*Open System Interconnection*), fue diseñado desde sus inicios para dar solución a los problemas de comunicación que se daban entre los equipos de cómputo de diferentes fabricantes, por lo que la Organización Internacional de Normalización, ISO por sus siglas en inglés (*International Organization for Standardization),* que es encargada de realizar normas internacionales, en 1984 lanza este modelo de red, con lo que vino a dar un orden al gran crecimiento que se estaba dando en las redes, y que se siguen expandiendo a gran velocidad el día de hoy en conjunto a las nuevas tecnologías de red.

Así para enfrentar el problema de incompatibilidad de las redes la ISO, propuso una serie de reglas generales a todas las redes. Con esto se cumplían objetivos como una estructura multinivel, la cual está hecha con la idea de que cada nivel se dedique a resolver un problema de comunicación y a cumplir una función específica. Entre los diferentes niveles existen interfases llamadas puntos de acceso, los niveles tienen una dependencia del nivel inferior como del superior, y una serie de mensajes de control lo que incorpora una robustez al diseño de este modelo.

El modelo hace comprender que existe un gran potencial en todo sistema de cómputo o telecomunicación pero que de verdad cobra importancia en el momento de transmitir datos y éstos llegan a su destino.

El modelo OSI consta de 7 niveles o capas, como se muestra en la figura 1.

Nivel de Aplicación **Servicios de red a aplicaciones**
Nivel de Presentación **Representación de los datos**
Nivel de Sesión **Comunicación entre dispositivos de la red**
Nivel de Transporte **Conexión extremo-extremo y fiabilidad de los datos**
Nivel de Red **Determinación de ruta e IP (Direccionamiento lógico)**
Nivel de Enlace de Datos **Direccionamiento físico**
Nivel Físico **Señal y transmisión binaria**

Figura 1 Pila del modelo OSI.

- **Capa Física.** Es el primer nivel de los siete del modelo OSI, ésta es la capa más baja, y se encarga de las conexiones físicas y mecánicas de los dispositivos, de cuidar su correcta operación, trata todo lo referente al medio físico como el cable coaxial, cable de par trenzado, fibra óptica y otros tipos de cables, así como los medios no guiados, como son radio, infrarrojo, microondas y redes inalámbricas.

  También se encarga de la transmisión de los bits de información a través del medio, tiene que preocuparse por las propiedades físicas y características eléctricas de los diversos componentes, de la velocidad de transmisión y si ésta es unidireccional o bidireccional. Se ocupa de transformar una trama de datos proveniente del nivel de enlace en una señal adecuada al medio físico utilizado en la transmisión y poderlo entregar como impulsos eléctricos, por lo que de esta misma forma recibirá impulsos eléctricos y los transformará para entregarlos como una trama de datos al nivel de enlace.

- **Capa de Enlace.** Es el segundo nivel del modelo, se encarga de proporcionar una transmisión sin errores, es decir, un tránsito de datos fiable a través de un enlace físico, ya que esta capa consigue que la información fluya libre de errores entre dos equipos conectados a través de un medio. Esto lo hace mediante tramas que se utilizan como unidades de información de sentido lógico para realizar el intercambio de información por medio de la capa de enlace. Por lo que se puede decir que esta capa realiza las siguientes funciones:

  - ***Iniciación*.-** Comprende los procesos necesarios para la activación del enlace e implica el intercambio de tramas de control, con el fin de saber la disponibilidad del medio, para el caso de transmitir o recibir información.
  - ***Terminación.-*** Cumple la función de liberar los recursos ocupados hasta la recepción y envío de la última trama.
  - ***Segmentación.-*** Comprende el envío de tramas muy largas, ya que si una trama es muy extensa tienen que dividirse en tramas más pequeñas, por lo que se tienen varias tramas de un cierto tamaño para aumentar la eficiencia de envío. Así también se encuentra el caso de las tramas que son demasiado cortas, ya que éstas pueden agruparse en una sola trama más larga para su envío.
  - ***Sincronización.-*** Comprende los procesos necesarios para adquirir, mantener y recuperar la sincronización de caracter u octeto, es decir, poner en fases los mecanismos de codificación del emisor con los mecanismos de decodificación del receptor. Ya que en la transferencia de información, en el nivel de enlace es necesario identificar los bits y saber que posición les corresponde en cada caracter u octeto dentro de una serie de bits recibidos.
  - ***Delimitación de la trama.-*** Es efectuada por varios métodos ya que en esta capa además de encargarse de la sincronización, por medio del método de caracter de

principio y fin, el método guión y el de principio, se puede tener un control del tamaño de las tramas.

- ***Control de errores.-*** Proporciona detección y corrección de errores en el envío de tramas entre las computadoras, y con esto puede ejercer el control de la capa física. Los métodos más comunes que se pueden encontrar para el control de errores son el FEC o corrección de errores por anticipo, que no tiene control de flujo, el control de flujo, ARQ, por sus siglas en inglés (*Automatic Repeat Request*) que lo realiza mediante parada y espera o ventana deslizante.
  La detección de errores se realiza por diferentes tipos de códigos como el control de redundancia cíclica, CRC, por sus siglas en inglés (*Cyclic Redundancy Check*), bit de paridad, paridad cruzada, Checksum.

- ***Control de flujo.-*** Esto es necesario para no saturar al receptor, este proceso se realiza en el nivel de transporte pero también puede darse en el nivel de enlace, utilizando mecanismos de retroalimentación que suelen ir unidos a la corrección de errores y no deben limitar la eficiencia del medio.

- **Capa de Red**. Es la que proporciona conectividad entre dos redes que aunque no se encuentren conectadas directamente, puede llevar paquetes desde un origen a un destino, esto lo hace a través de algoritmos de encaminamiento. El funcionamiento de ésta puede darse a través de datagramas que son paquetes que se encaminan independientemente, sin que el origen y el destino tengan que pasar por un establecimiento previo de comunicación, o los circuitos virtuales, que en una red de circuitos virtuales, donde dos equipos que quieran comunicarse tengan que empezar por establecer una conexión.

  En este nivel los dispositivos esenciales son los ruteadores, ya que gracias a ellos puede darse la conexión entre redes, así como poder encontrar el mejor camino hacia las mismas, en esta capa también se encuentran los servicios orientados a conexión y orientados a la no conexión.

  Orientados a conexión, en éstos los paquetes llevan una secuencia ordenada, cada datagrama debe llevar su dirección destino, ya que con esto se puede establecer la mejor ruta que seguirán los paquetes pertenecientes a la conexión.

  Orientados a no conexión, los paquetes llevan dirección destino, pero no tienen establecida la mejor ruta, por lo que cada uno decide el camino que seguirá.

- **Capa de Transporte.** Su función es recibir los datos de las capas superiores, dividirlos si es necesario y enviarlos a la capa de red, con esto se establece la conexión, no importando que tipo de tecnología se aplique para la conexión o construcción de la red, en la que se encuentre el host emisor y el receptor.

Para las redes de comunicación existen dos principales protocolos de transporte y se dividen en: orientado a conexión y a la no conexión. El protocolo orientado a la no conexión es UDP, mientras que el orientado a conexión es TCP.

- **Capa de Sesión.** Tiene como principal función establecer, administrar y finalizar una sesión entre dos hosts, ésta presta sus servicios a la capa de presentación ya que proporciona un control de sesión, control de concurrencias y mantiene puntos de verificación. Por lo que se concluye que la capa asegura, que dada una sesión establecida entre dos máquinas se pueda dar de principio a fin y reanudándola en caso de interrupción.

- **Capa de Presentación.** Es la encargada de garantizar que la información que envía la capa de aplicación de un sistema pueda ser leída por la capa de aplicación de otra. Ya que presentar los datos de forma legible al sistema receptor, se encarga de la forma y estructura de los datos para que puedan ser interpretados de forma comprensible por los diferentes sistemas.

- **Capa de aplicación**. Es la que trabaja más cerca a los usuarios, ya que ofrece a las aplicaciones que se están ejecutando, acceder a los servicios de las demás capas para intercambiar información, seleccionando los protocolos adecuados para realizar las distintas tareas como, el intercambio de correos electrónicos (POP y SMTP), gestores de datos, entre otros.

  Aunque ésta es la más cercana al usuario no quiere decir que los usuarios interactúen directamente con ella, dado que los usuarios están interactuando directamente con la aplicación, y ésta a su vez es la que interactúa con este nivel, haciendo invisible este proceso al usuario.

  Algunos de los protocolos de aplicación más conocidos para esta capa son:

  - DNS (Domain Name System).
  - FTP (File Transfer Protocol) (FTAM, fuera de TCP/IP).- para transferencia de archivos.
  - HTTP (HyperText Transfer Protocol).- el protocolo bajo la www.
  - POP (Post Office Protocol)/IMAP.- para el reparto de correo al usuario final.
  - SMTP (Simple Mail Transfer Protocol) (X.400 fuera de TCP/IP).- para el envío y distribución de correo electrónico
  - SNMP (Simple Network Management Protocol).- Para facilitar el intercambio de información de administración entre dispositivos de red.
  - SSH (Secure SHell).- Principalmente para terminal remota, cifrando casi cualquier tipo de transmisión.

- Telnet (terminal remota) .- Aunque ha caído en desuso por su inseguridad intrínseca, ya que las claves viajan sin cifrar por la red, aún se utiliza.

1.2 Pila de protocolos TCP/IP

La familia de protocolos de Internet son un conjunto de reglas que especifican el intercambio de datos u órdenes durante la comunicación entre los diferentes puntos de la red, ya que permiten el flujo de información entre computadoras que manejan lenguajes distintos, dos computadoras conectadas en una red no podrían comunicarse si no "hablaran" el mismo idioma, por tal motivo, en Internet los protocolos que fueron acogidos para que las comunicaciones pudieran darse entre equipos son los protocolos TCP/IP. Éstos fueron adoptados gracias a que los protocolos TCP/IP, en referencia a los dos protocolos más importantes que lo componen, el Protocolo Control de Transmisión (TCP) y Protocolo de Internet (IP) y ahora son los protocolos de comunicación más utilizados.

La pila TCP/IP es la base del Internet actualmente, y sirve para enlazar computadoras que utilizan diferentes sistemas operativos, como PCs, minicomputadoras y computadoras centrales sobre redes de área local y áreas extensas. TCP/IP fue desarrollado y utilizado por primera vez en 1972 por el departamento de defensa de los Estados Unidos de Norteamérica, ejecutándolo en ARPANET, una red de área extensa de este departamento.
El formato del encabezado TCP se describe en la figura 1.2

Bits	**0 – 3**	**4 – 7**	**8 - 15**	**16 – 31**
0	Puerto Origen			Puerto Destino
32	Número de Secuencia			
64	Número de Acuse de Recibo (ACK)			
96	Longitud del encabezado TCP	Reservado	Banderas	Ventana
128	Suma de Verificación (Checksum)			Puntero Urgente
160	Opciones + Relleno (opcional)			
224	Datos			

Figura 1.2 Encabezado de TCP

Descripción de los campos:

- *Puerto de origen* (16 bits): Identifica el puerto origen.
- *Puerto destino* (16 bits): Identifica el puerto destino.
- *Número de secuencia* (32 bits): Sirve para comprobar que ningún segmento se ha perdido, y que llegan en el orden correcto. Su significado varía dependiendo del valor de SYN, que pueden ser:
    - Si la bandera SYN está activa (1), entonces este campo indica el número inicial de secuencia (con lo cual el número de secuencia del primer byte de datos será este número de secuencia más uno).
    - Si la bandera SYN no está activa(0), entonces este campo indica el número de secuencia del primer byte de datos.
- *Número de acuse de recibo* (ACK) (32 bits): Si la bandera ACK está activa, entonces este campo contiene el número de secuencia del siguiente byte que el receptor espera recibir.
- *Longitud del encabezado TCP* (4 bits): Especifica el tamaño de la cabera TCP en palabras de 32-bits. El tamaño mínimo es de 5 palabras y el máximo es de 15 palabras (lo cual equivale a un tamaño mínimo de 20 bytes y a un máximo de 60 bytes). En inglés el campo se denomina “Data offset”, que literalmente significa “desplazamiento hasta los datos”, ya que indica cuántos bytes hay entre el inicio del paquete TCP y el inicio de los datos.
- *Reservado* (4 bits): Bits reservados para uso futuro, deberían ser puestos a cero.
- *Banderas* (8 bits): Son 8 banderas. Cada una se indica con un 1 si está “activa” o “inactiva” con un 0. Las diferentes banderas se describen a continuación:
    - CWR o “Congestion Window Reduced” (1 bit): Esta Bandera se activa (se pone a 1) por parte del emisor para indicar que ha recibido un paquete TCP con la bandera ECE activada. ECE es una extensión del protocolo que fue añadida al encabezado en el RFC 3168 [1]. Se utiliza para el control de la congestión en la red.
    - ECE o “ECN-Echo” (1 bit): Indica que el receptor puede realizar notificaciones ECN. La activación de esta bandera se realiza durante la negociación en tres pasos para el establecimiento de la conexión. Esta bandera también fue añadida al encabezado en el RFC 3168 [1].
    - URG o “urgent” (1 bit): Si está activa, significa que el campo “Urgente” es significativo, sino, el valor de este campo es ignorado.
    - ACK o “acknowledge” (1 bit): Si está activa, entonces el campo con el número de acuse de recibo es válido (sino, es ignorado).
    - PSH o “push” (1 bit): Activa/desactiva la función que hace que los datos de ese segmento y los datos que hayan sido almacenados anteriormente en el buffer del receptor deben ser transferidos a la aplicación receptora lo antes posible.
    - RST o “reset” (1 bit): Si llega a 1, se reinicia la conexión.
    - SYN o “synchronize” (1 bit): Activa/desactiva la sincronización de los números de secuencia.
    - FIN (1 bit): Si se activa es porque no hay más datos a enviar por parte del emisor, esto es, el paquete que lo lleva activo es el último de una conexión.

- *Ventana* (16 bits): Es el tamaño de la ventana de recepción, que especifica el número de bytes que el receptor está actualmente esperando recibir.
- *Suma de verificación* (checksum) (16 bits): Es una suma de verificación utilizada para comprobar si hay errores tanto en el encabezado como en los datos.
- *Puntero urgente* (16 bits): Si la bandera URG está activa, entonces este campo indica el desplazamiento respecto al número de secuencia que indica el último byte de datos marcados como "urgentes".
- *Opciones* (número de bits variable): La longitud total del campo de opciones ha de ser múltiplo de una palabra de 32 bits (si es menor, se ha de rellenar al múltiplo más cercano), y el campo que indica la longitud del encabezado ha de estar ajustado de forma adecuada.
- *Datos* (número de bits variable): No forma parte del encabezado, es la carga (payload), la parte con los datos del paquete TCP. Pueden ser datos de cualquier protocolo de nivel superior en el nivel de aplicación; los protocolos de aplicación más comunes para los que se usan los datos de un paquete TCP son HTTP, Telnet, SSH, FTP, etcétera.

El protocolo de datagrama del usuario (UDP).- Es un protocolo orientado a la no conexión, que proporciona un servicio no confiable de datagrama, no proporciona ninguna detección o corrección extremo a extremo, no retransmite ningún dato no recibido y no tiene ninguna capacidad para efectuar control de error o de flujo. Es por eso que todos los programas de aplicación con base a UDP tienen la responsabilidad de cubrir con mecanismos para el control de flujo y error, y la recuperación de paquetes perdidos por las carencias del protocolo. Así que la confiabilidad de los datos recae en los programas, esto hace que UDP sea más rápido en su desempeño cuando la red no está congestionada, porque lleva menos sobre carga que TCP. Sin embargo, cuando la red está congestionada, las aplicaciones con base en UDP muy probablemente resultarán en sesiones de tiempo agotado y bajo desempeño. La forma de la trama UDP, se muestra en la figura 1.3.

0 – 15 bits	16 – 31 bits
Puerto origen	Puerto destino
Longitud	Suma de Verificación (Checksum)
Datos	

Figura 1.3 Trama UDP.

Descripción de los campos:

- *Puerto origen* (16 bits): Número de puerto o servicio (aplicación) en el sistema origen.
- *Puerto destino* (16 bits): Número de puerto o servicio (aplicación) en el sistema destino.
- *Longitud* (16 bits): longitud en bytes del datagrama, incluyendo el encabezado UDP y los datos. La longitud mínima es de 8 bytes.

- *Suma de verificación* (checksum) (16 bits): Es un campo de comprobación de la integridad del encabezado UDP, el pseudo encabezado y los datos. Si el valor es cero indica que no debe realizarse ningún cálculo relativo al checksum.

El protocolo de Internet (IP).- Es el protocolo principal de TCP/IP, encargado de la transmisión y enrutamiento de los paquetes de datos al equipo destino. Es un protocolo no fiable o también llamado *el del mejor esfuerzo*, es decir, que no asegura la recepción fina de la información en el equipo destinatario y únicamente proporciona seguridad mediante *checksum* o sumas de comprobación de los encabezados y no de los datos transmitidos. Para el control de los posibles errores dispone del Protocolo de Mensajes de Control de Internet, ICMP, por sus siglas en inglés, (Internet Control Message Protocol), por lo que la fiabilidad de la comunicación la deben proporcionar los protocolos superiores, como TCP.

Los datos en una red basada en IP son enviados en bloques conocidos como paquetes o datagramas. Si la información a transmitir (datagramas) supera el tamaño máximo establecido, MTU por sus siglas en inglés, (Maximum Transfer Unit) en el segmento de red por el que va a circular, podrá ser fragmentada en paquetes más pequeños, y reensamblada luego cuando sea necesario. Estos fragmentos podrán ir cada uno por un camino diferente dependiendo de como estén congestionadas las rutas, para así llegar a su destino. El formato del encabezado IP se muestra en la figura 1.4

0 – 3 bits	4 – 7 bits	8 – 15 bits	16 – 31 bits
Versión	IHT	Tipo de Servicio	Longitud total
Identificación		Banderas	Desplazamiento
TTL		Protocolo	Checksum
Dirección Origen			
Dirección destino			
Opciones			Relleno
Datos			

Figura 1.4 Formato del encabezado IP.

Descripción de los campos:

- *Versión* (4 bits): el campo de versión especifica el formato del encabezado IP. Actualmente sólo se manejan dos, el IP estándar o versión 4 y la siguiente generación o versión 6(IPv6 o IPng)
- *Longitud del encabezado, IHL* por sus siglas en inglés (Internet Header Length) (4 bits): longitud del encabezado IP medida en palabras de 32 bits. La longitud mínima es de 5 palabras (encabezado sin opciones IP).
- *Tipo de servicio* (8 bits): se emplea para especificar parámetros como la fiabilidad, la precedencia, el retardo y la capacidad de salida que deberían asociarse a este paquete.
- *Longitud total* (16 bits): longitud total del datagrama IP en bytes, incluyendo el encabezado IP y los datos encapsulados por éste.
- *Identificación* (16 bits): este campo identifica de forma unívoca cada paquete enviado por un sistema emisor. Es empleado para la reconstrucción de paquetes grandes que debieron fragmentarse en algún punto de la red.
- *Banderas* (3bits):
    - El primer bit, (More Flag, MF), se emplea en la fragmentación, para determinar si este paquete es el último fragmento de un paquete inicial, o si aún siguen más fragmentos.
    - El segundo bit, (Don't Fragment, DF), permite especificar si el emisor desea que se fragmente el datagrama.
    - El último bit de esta bandera actualmente no es utilizado.
- *Desplazamiento* (13 bits): Mediante el offset o desplazamiento es posible determinar la posición que ocupaba en el paquete original este fragmento, de forma que el host destino pueda reconstruirlo consecuentemente.
- *Tiempo de vida* (8 bits): El TTL, por sus siglas en inglés (Time to live), indica el tiempo máximo que el paquete puede permanecer en la red. Si su valor es cero, el paquete es destruido. Típicamente, en lugar de indicar un valor temporal se refiere al número de ruteadores, o saltos, por los que puede pasar.
- *Protocolo* (8 bits): Especifica el protocolo de siguiente nivel empleado, que recibirá los datos en el otro extremo. Es decir, el contenido del campo de datos comenzará por el encabezado del siguiente protocolo, por ejemplo TCP o UDP.
- *Suma de Verificación (Checksum)* (16 bits): Es un campo de comprobación de la integridad del encabezado IP. Debido a que ciertos campos del encabezado cambian en el transcurso del paquete por la red, por ejemplo el TTL en cada salto, éste campo debe recalcularse y verificarse en cada punto intermedio dónde el encabezado es procesado.
- *Dirección origen* (32 bits): Dirección IP del dispositivo fuente o emisor.
- *Dirección destino* (32 bits): Dirección IP del dispositivo destino o receptor.

- *Opciones* (variable): La longitud de este campo es variable, pudiendo no tener opciones o más de una opción. Concretamente contempla las opciones solicitadas por el emisor, que pueden ser de seguridad, "source routing", timestamps.
- *Relleno* (variable): Asegura que el encabezado IP acaba en un múltiplo de 32 bits.
- *Datos* (variable): Este campo contiene los datos a enviar, siendo su longitud múltiplo de 8 bits. El valor máximo de la longitud es 65.535 bytes (64 Kbytes). El campo comenzará con el contenido del encabezado del protocolo de siguiente nivel: TCP o UDP.

La pila de protocolos TCP/IP consta de 4 capas y en comparación con el modelo OSI, las tres capas superiores (Aplicación, Presentación y Sesión) son consideradas como el nivel de aplicación en la pila TCP, como se muestra en la figura 1.5. TCP/IP no tiene un nivel de sesión unificado sobre el que los niveles superiores se sostengan, estás funciones son típicamente desempeñadas por las aplicaciones de usuario. Está es la diferencia más notable entre los modelos de TCP/IP y el modelo OSI, ya que en el nivel de Aplicación en TCP/IP se integran algunos niveles del modelo OSI.

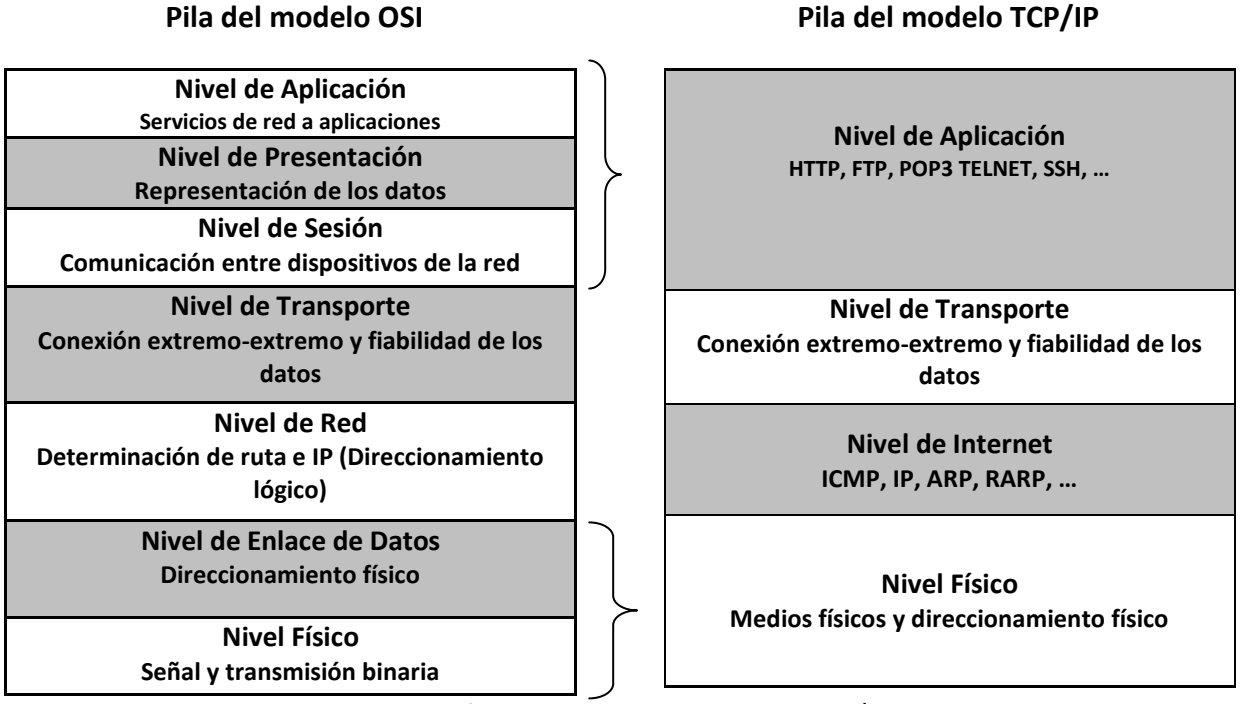


Figura 1.5 Comparación de modelos OSI contra el TCP/IP

- ***Capa o nivel físico***. Es muy similar a la capa del modelo OSI ya que se describen las características físicas de las comunicaciones, sobre las características del medio a utilizar y todo lo relativo a los detalles de los conectores, código de canales y modulación, potencia de la señal entre otros, para que pueda darse la transmisión binaria.

- ***Capa o nivel de Internet.*** Es la que se encarga del envío y recepción de los paquetes a través de la red, así como de encaminarlos por diferentes rutas que deben recorrer para llegar a su destino. Principalmente IP junto a ICMP se encargan de esa tarea.

- ***Capa o nivel de transporte.*** Se encarga de manejar los flujos de datos entre equipos. Los protocolos que más se utilizan para realizar esta tarea son: TCP ya que es un protocolo confiable y que está orientado a conexión, y UDP un protocolo simple que como se había mencionado tiene desventajas contra TCP, utilizado por brindar un buen desempeño bajo ciertas condiciones de trabajo.

- ***Capa o nivel de aplicación.*** Es el nivel donde programas más comunes utilizan para comunicarse a través de una red con otros programas. Los procesos que acontecen en este nivel, son aplicaciones específicas que pasan los datos al nivel de aplicación en el formato que internamente use el programa y es codificado de acuerdo con un protocolo estándar.
  Algunos programas específicos se considera que se ejecutan en este nivel. Proporcionan servicios que directamente trabajan con las aplicaciones de usuario.
  Estos programas y sus correspondientes protocolos incluyen a HTTP (HyperText Transfer Protocol), FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), DNS (Domain Name System) entre otros.

Por lo que el objetivo de establecer normas y reglas por comités e instituciones reguladoras, es para que cuando se realice la transmisión de algún dato, por algún sistema de red, este intercambio entre los dispositivos, que se da por algún medio de comunicación, pueda realizarse siguiendo estos lineamientos que se han establecidos por el consenso y participación de todos los involucrados, y así se tenga como resultado sistemas de comunicación conformados por software y hardware que persigan un bien común para el usuario, y se puedan realizar transmisiones eficientes y confiables.

Capítulo 2
Protocolo de Internet versión 6: IPv6

2.1. Introducción.

El modelo TCP/IP fue creado hace más de 25 años, con el cual se ha podido avanzar en el desarrollo de nuevas tecnologías, por su flexibilidad y múltiples usos, este conjunto de protocolos se encarga de transportar paquetes de información de un equipo a otro a través de las redes de datos, es posible darse cuenta que su trabajo sobre la capa de red y transporte es esencial e importante, por esto es que las redes actuales trabajan con el protocolo de Internet versión cuatro IPv4, por sus siglas en inglés.

IPv4 con el paso del tiempo ha presentado limitaciones para dar un buen funcionamiento a las redes ya que éstas han ido evolucionando, por lo que el protocolo no presenta un futuro alentador para las redes futuras. La demanda de direcciones, por ejemplo, como lo hace ver el Registro de Direcciones de Internet para América Latina y el Caribe , LACNIC por sus siglas en inglés (*Latin American and Caribbean Internet Addresses Registry*), hace una comparación global del número de direcciones IPv4 asignadas hasta el 12 de agosto de 2008 para cada Registro de Internet Regional, RIR por sus siglas en inglés *(Regional Internet Registry*), tomando en cuenta las direcciones legadas, es decir, aquellas asignadas por el Registro Central antes de la creación de los RIR's; presentando del total de 152.26 bloques de /8 asignados hasta el momento, a la entidad de registro de África, AfriNIC por sus siglas en inglés (African *Network Information Center*) con 1.15, a la entidad de Registro de Asia, APNIC por sus siglas en inglés (*Asia Pacific Network Information Center*) con 27.93, a la entidad de Registro de Norteamérica, ARIN por sus siglas en inglés (*American Registry for Internet Numbers*) con 88.72, la entidad de registro de América Latina y el Caribe, LACNIC con 4.40 y a la entidad de Registro de Europa, RIPENCC por sus siglas en inglés (*Réseaux IP Européens Network Coordination Centre*) con 30.06, siendo que cada bloque de prefijo /8 contiene 16777216 direcciones IPv4. Así se muestra un panorama en el cual, el número de direcciones disponibles es cada vez menor, por lo que el grupo de trabajo de la IETF por sus siglas en inglés (Internet Engineering Task Force), llamado ALE, por sus siglas en inglés (Address Lifetime Expectations), realizó un estudio para hacer una estimación sobre la expectativa de vida de IPv4 y el resultado de su investigación dio como resultado que las direcciones existentes se terminarían de delegar entre el 2005 y 2011, aunque este tiempo podría variar.

Adicionalmente, hoy en día el ruteo es ineficiente, por lo que se han ideado algunas formas para alargar el tiempo de vida de IPv4, como es el caso de los traductores de direcciones de red, NAT por sus siglas en inglés (Network Address Translation), que es un mecanismo que genera una solución al problema de escasez al proveer mayor número de direcciones IPv4 pero también genera retos para las nuevas tecnologías que se desarrollan, por lo que el soporte para nuevas aplicaciones es inadecuado, ya que las mismas son más demandantes,

requieren garantías en tiempos de respuesta, disponibilidad de ancho de banda y/o Calidad de Servicio, QoS por sus siglas en inglés (Quality of Service) entre otros, lo que hace más difícil a IPv4 adecuarlo a las nuevas aplicaciones.

En IPv4 la seguridad es opcional, ya que éste no fue diseñado para ser seguro, debido a que originalmente sólo se utilizaba para el uso exclusivo de redes militares, de investigación y educación que eran entornos aislados, por lo que se excluyeron varias opciones que hacían que está versión del protocolo fuera inherentemente insegura. Posteriormente las redes aisladas se convirtieron en una red pública para fines comerciales y ahora es posible ver las vulnerabilidades y carencias del mismo.

El esfuerzo para desarrollar un protocolo nuevo o una versión mejorada que sucediera a IPv4 comenzó a principios de 1990 por la IETF, ya que se consideraba necesario que diera solución a algunos de los problemas de diseño de IPv4 y el más importante por resolver, que es y ha sido el problema del espacio de direcciones que empezaba a presentar la versión 4 del protocolo de Internet, para que así el sucesor fuese más funcional, robusto y seguro. Por lo que en 1993 la IETF comenzó con una nueva área de investigación llamada "Protocolo de Internet de siguiente generación", IPng, por sus siglas en inglés ("Internet Protocol Next Generation"), para estudiar diferentes propuestas y recomendaciones para el desarrollo de un nuevo protocolo o versión del actual.

El director del área del IPng recomendó la creación de IPv6 en la reunión de la IETF de 1994 en Toronto. Esa recomendación está especificada en el RFC 1752 [2]. Ese mismo año el grupo de investigación del IPng dio a conocer los criterios técnicos en el RFC 1726 [3], a cumplir para escoger una nueva versión del protocolo de Internet, un conjunto de 17 criterios, que se dieron a conocer, para que cumpliera el nuevo protocolo o versión, fueron los siguientes:

- Escalabilidad: El nuevo protocolo debería ser capaz de identificar y direccionar por lo menos 10^{12} sistemas finales y 10^{9} redes individuales.
- Flexibilidad topológica: La arquitectura de enrutamiento y protocolos para IPng debían permitir utilizar muchas topologías distintas de red.
- Rendimiento: Para IPng los host deberían ser capaces de transferir datos a tasas comparables a las alcanzadas con IPv4, utilizando niveles similares de recursos máquina.
- Servicio robusto: El servicio de red junto con los protocolos de control y enrutamiento para IPng deberían ser suficientemente robustos.
- Transición: Debían existir mecanismos para realizar la transición de IPv4 hacia IPng de manera transparente para los protocolos y aplicaciones de las capas superiores.
- Independencia del medio: Este nuevo protocolo debía de trabajar a través de Internet por diferentes medios LAN, WAN y MAN, así como distintas velocidades de conexión, que van desde algunos bits/segundo hasta cientos de giga bits/segundo.

- Servicio de datagramas no confiables: En nuevo protocolo debía soportar un servicio no confiable de entrega de datagramas.
- Configuración, Operación y Administración: Este nuevo protocolo también debía permitir conexiones de una forma fácil, además de operación y configuración ampliamente distribuida. También debía permitir la configuración automática de hosts y enrutadores.
- Operación segura: IPng también debía proveer una capa de red segura (IPSec).
- Acceso y documentación: Los protocolos que definen a IPng, sus protocolos asociados y protocolos de enrutamiento deberían ser publicados en los RFC's, así como estar disponibles libremente y no requerir licencia para su implementación.
- Nombres únicos: IPng debía asignar a todos los objetos de la capa IP de manera global nombres de Internet únicos.
- Multicast: IPng debía soportar transmisión de paquetes Unicast y multicast.
- Extensibilidad: IPng debía ser capaz de evolucionar para cubrir las necesidades futuras de Internet. Así mismo, conforme éste evolucione, debería permitir diferentes versiones de él, que puedan coexistir sobre la misma red.
- Servicio de red: IPng debía permitirle a la red asociar paquetes con clases de servicio en particular y proveerlas con los servicios especificados por esas clases.
- Movilidad: El protocolo debía soportar huéspedes, redes fijas y redes móviles.
- Protocolo de control: El protocolo debía incluir soporte elemental para probar y depurar redes.
- Redes privadas: Por último, IPng debía permitir a los usuarios construir redes privadas sobre la infraestructura básica de red, soportando ambas, redes basadas o no basadas en IP

En base a los criterios se pudieron seleccionar varias propuestas, de las cuales CATNIP, TUBA y SIPP fueron las tres principales, después de una pequeña discusión que se pueden referir en el RFC 1752 [2], la propuesta que se aceptó fue la de SIPP, además de incorporarle direcciones de 128 bits de longitud y hacer algunas otras modificaciones. El resultado final de todas estas modificaciones es lo que se conoce actualmente como IPv6 ó IPng.

2.2. Características de IPv6.

IPv6 incluye mecanismos de transición e interoperabilidad los cuales están diseñados para que los usuarios adopten e implementen IPv6 paso a paso según se requiera y para proporcionar interoperabilidad entre hosts IPv4 e IPv6.

El cambio de IPv4 a IPv6 recae principalmente en las siguientes categorías:

- Extendiendo las capacidades de direccionamiento (Expanded Addressing Capabilities).

  IPv6 aumenta el tamaño de la dirección IP de 32 bits a 128 bits, para soportar más niveles de direccionamiento jerárquico, un número muy grande de nodos direccionables y una simple autoconfiguración de direcciones. La escalabilidad de

ruteo multicast es una mejora al agregar un campo de alcance (scope) a la dirección multicast.

- Simplificación del formato de encabezado (Header Format Simplification).
  Algunos campos del encabezado de IPv4 se han desechado o vuelto opcionales así se pueden reducir el costo de los procesos y se limita la manipulación del ancho de banda y costo del encabezado IPv6.
- Mejora el soporte para extensiones y opciones (Improved Support for Extensions and Options).
  Cambia la forma en que el campo de opciones de IP está codificado y permite una mejor eficiencia en la transmisión, menos estrictos en los límites de la longitud de las opciones y mayor flexibilidad para introducir nuevas opciones en el futuro.
- Capacidad en el flujo de etiquetado (Flow Labeling Capability).
  Es agregada para habilitar el etiquetado de paquetes pertenecientes a un tráfico en particular llamado flujos, para que el remitente de las solicitudes tenga un manejo especial, no tiene por defecto la calidad de servicio o servicio en tiempo real.
- Autenticación y privacidad de capacidades (Authentication and Privacy Capabilities).
  Extensiones para apoyar la autenticación, integridad de datos y (opcionalmente) confidencialidad de datos, están especificados para IPv6.

Se tomaron en cuenta muchos aspectos para el diseño de IPv6, ya que se puso a discusión el uso de las direcciones fijas a 64 bits contra el direccionamiento de longitud variable de más de 160 bits. Finalmente utilizando direcciones de longitud fija de 128 bits se pudo ver que era la opción más adecuada. Con IPv4, el número total de nodos posibles en teoría es de 4,294,967,296 (2^{32}), lo que representa cerca de menos de dos direcciones por cada tres personas (basándonos en una población mundial de 6,706,992,932 de personas en el 2008). Mientras tanto, con la longitud de 128 bits de IPv6, que representa $3.4(10^{38})$ direcciones, permite aproximadamente $5.07(10^{28})$ direcciones IPv6 para cada persona en el mundo.

No obstante, como sucede en cualquier esquema de direccionamiento, no todas las direcciones pueden ser usadas, pero las direcciones restantes es un número más que suficiente, y está disponible para cualquier tipo de uso. Incrementar el número de bits por dirección también significa incrementar el tamaño del encabezado IP. Debido a que cada cabecera IP contiene la dirección de origen y destino, el tamaño del campo de direcciones para IPv4 es de 64 bits y para IPv6 es de 256 bits.

Al poder hacer una comparación de las principales diferencias realizadas en el modelo OSI con IPv4 y aquellas con IPv6, da como resultado el presentado en la figura 2.1, donde se representa únicamente un cambio en la capa de red, esto representa un punto fundamental cuando se desarrolló IPv6. Las otras capas de los dos modelos de referencia OSI permanecen igual, lo cual significa que protocolos como TCP y UDP, continúan

funcionando de forma normal, y no sufrieron grandes alteraciones por el cambio de versión del protocolo.

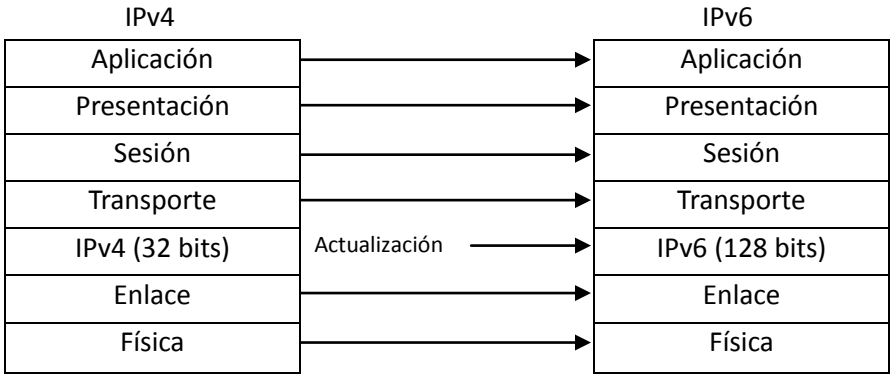


Figura 2.1. Cambios de IPv4 e IPv6 con respecto al modelo OSI.

2.2.1. Representación de las direcciones de IPv6.

Dado que las direcciones IPv6 por su longitud deben de tener una forma especial para su representación, se han ideado varias técnicas para poder presentar de forma sencilla los 128 bits de cada dirección.

Una de las formas en las cuales se puede expresar una dirección IPv6 es la notación completa, como se muestra en la tabla 2.1, ya que con esta notación los 128 bits de la dirección IPv6, se expresan mediante 8 bloques de 16 bits, cada uno de estos separados mediante el signo dos puntos ":". Los 16 bits de cada bloque se traducen a cuatro caracteres hexadecimales, por lo tanto, un bloque puede tener valores hexadecimales que van desde 0x0000 hasta 0xFFFF.

Representación en notación completa de direcciones IPv6.
0000:0000:0000:0000:0000:0000:0000:0000
2001:1218:0001:0006:FFFF:5000:4343:AAAA
3FFF:0002:4343:1234:12FF:FE42:1111:2222

Tabla 2.1 Ejemplo de notación completa de las direcciones IPv6.

Otra técnica para representar las direcciones IPv6, que es de uso muy común, es la notación simplificada, debido a que es de gran ayuda para las direcciones que contienen largas cadenas de ceros, es por esto, que se creó una sintaxis especial que simplifica valores consecutivos de ceros, y con ellos se definieron dos criterios:

- Bloques consecutivos de ceros.
- Bloques que comienzan con uno o más ceros.

La simplificación de las direcciones IPv6 que presentan **bloques consecutivos de ceros**, se realiza al sustituir estos bloques por un par de dos puntos "::". En una dirección IPv6, se permite el uso del par de dos puntos una sola vez. Cuando una dirección IPv6 presenta este par de dos puntos "::", un analizador de dirección debe identificar el número de ceros omitidos. Cuando el analizador de dirección identifica la cantidad de ceros sustituídos por el signo "::", éste llena con ceros el especio entre los dos puntos de la dirección hasta que se completan los 128 bits de la dirección, como se muestra en la tabla 2.2.

Representación en notación completa de direcciones IPv6.	Representación en notación simplificada de direcciones IPv6.
0000:0000:0000:0000:0000:0000:0000:0000	::
2001:0000:0000:0000:FFFF:0000:4343:AAAA	2001::FFFF:0000:4343:AAAA
FE80:0000:0000:0000:0000:0000:0000:2222	FE80::2222

Tabla 2.2 Ejemplo de representación de bloques consecutivos de ceros.

La simplificación de **bloques que comienzan con uno o más ceros**, se puede realizar eliminando éstos para su simplificación, sin embargo, si todo el bloque contiene ceros, al menos un carácter 0 se debe mantener, como se muestra en la tabla 2.3.

Representación en notación completa de direcciones IPv6.	Representación en notación simplificada de direcciones IPv6.
0000:0000:0000:0000:0000:0000:0000:0000	0:0:0:0:0:0:0:0
2001:0000:0000:0000:FFFF:0000:4343:AAAA	2001:0:0:0:FFFF:0:4343:AAAA
FE80:0000:0000:0000:0000:0000:0000:222	FE80:0:0:0:0:0:0:222

Tabla 2.3 Ejemplo de representación de bloques que comienzan con uno o más ceros.

Otras opciones que nos ofrece esta nueva versión del protocolo, es la combinación de métodos de simplificación, por lo que al tener varios bloques de 0´s consecutivos y de 0´s que inician cada bloque, se pueden combinar, como se muestra en la tabla 2.4.

Representación en notación completa de direcciones IPv6.	Representación en notación simplificada de direcciones IPv6.
2001:0123:BF00:AAAA:0000:0000:0002:5454	2001:123:BF00:AAAA::2:5454
2001:0000:0000:0000:0FFF:0001:0343:0AAA	2001::FFF:1:343:AAA
FE80:0000:0000:0000:0000:0000:0000:0022	FE80::22

Tabla 2.4 Ejemplo de combinación de métodos de simplificación de direcciones IPv6.

2.2.2. Encabezado de IPv4.

En la actualidad las redes de Internet tienen como base de su funcionamiento el protocolo IP, que se puede encontrar especificado en los RFC 791 [4], 950 [5] y 922 [6], al ser un protocolo que ha ayudado en el desarrollo de muchas tecnologías de comunicación en la actualidad a pesar de sus limitantes. El encabezado de IPv4 se describe en la figura 2.2.

<table>
<tr><td>0 – 3 bits</td><td>4 – 7 bits</td><td>8 – 15 bits</td><td colspan="2">16 – 31 bits</td></tr>
<tr><td>Versión</td><td>IHT</td><td>Tipo de Servicio</td><td colspan="2">Longitud total</td></tr>
<tr><td colspan="3">Identificación</td><td>Banderas</td><td>Desplazamiento</td></tr>
<tr><td colspan="2">TTL</td><td>Protocolo</td><td colspan="2">Suma de verificación (Checksum)</td></tr>
<tr><td colspan="5">Dirección Origen</td></tr>
<tr><td colspan="5">Dirección destino</td></tr>
<tr><td colspan="4">Opciones</td><td>Relleno</td></tr>
<tr><td colspan="5">Datos</td></tr>
</table>

Figura 2.2. Formado del encabezado IPv4.

- Versión (4 bits): El campo de versión especifica el formato de la cabecera IP. Actualmente sólo se manejan dos, el IP estándar o versión 4 y la siguiente generación o versión 6(IPv6 ó IPng)
- Longitud del encabezado IHL, por sus siglas en inglés (Internet Header Length, 4 bits): Longitud del encabezado de Internet o IP, medido en palabras de 32 bits. La longitud mínima es de 5 palabras (sin opciones IP).
- Tipo de servicio TOS, por sus siglas en inglés (Type of Service, 8 bits): El TOS se emplea para especificar parámetros como la fiabilidad, la precedencia, el retardo y la capacidad de salida que deberían asociarse a este paquete.
- Longitud total (16 bits): Longitud total del datagrama IP en bytes, incluyendo el encabezado IP y los datos encapsulados por éste.
- Identificación (16 bits): Este campo identifica de forma unívoca cada paquete enviado por un sistema emisor. Es empleado para la reconstrucción de paquetes grandes que debieron fragmentarse en algún punto de la red.
- Banderas (3 bits):
    - El primer bit, Más fragmentos (More Flagment, MF), se emplea en la fragmentación, para determinar si éste paquete es el último fragmento de un paquete inicial, o si aún le siguen más fragmentos.
    - El segundo bit, No fragmento (Don't Fragment, DF), permite especificar si el emisor desea que se fragmente o no el datagrama.
    - El último bit o bandera actualmente no es utilizado.

- Desplazamiento (13 bits): Mediante el offset o desplazamiento es posible determinar la posición que ocupaba en el paquete original este fragmento, de forma que el destino pueda reconstruirlo consecuentemente.
- Tiempo de vida (8 bits): Indica el tiempo máximo que el paquete puede permanecer en la red. Si su valor es cero, el paquete es destruido. Típicamente, en lugar de indicar un valor temporal, se refiere al número de ruteadores o saltos por los que puede pasar.
- Protocolo (8 bits): Este campo especifica el protocolo de siguiente nivel empleado, que recibirá los datos en el otro extremo. Es decir, el contenido del campo de datos comenzará por la cabecera del siguiente protocolo, por ejemplo TCP o UDP.
- Suma de Verificación (Checksum, 16 bits): Es un campo de comprobación de la integridad, sólo del encabezado IP. Debido a que ciertos campos del encabezado cambian en el transcurso del paquete por la red, por ejemplo el tiempo de vida en cada salto, este campo debe recalcularse y verificarse en cada punto intermedio donde el encabezado es procesado.
- Dirección origen (32 bits): Dirección IP del dispositivo fuente o emisor.
- Dirección destino (32 bits): Dirección IP del dispositivo destino o receptor.
- Opciones (variable): La longitud del campo es variable, pudiendo tener cero o más opciones. Concretamente contempla las opciones solicitadas por el emisor, que pueden ser de seguridad, timestamps, etc.
- Relleno (variable): Asegura que el encabezado IP acaba en un múltiplo de 32 bits.
- Datos (variable): Este campo contiene los datos a enviar, siendo su longitud múltiplo de 8 bits. El valor máximo de la longitud es 65.535 bytes (64 Kbytes). El campo comenzará con el contenido del encabezado del protocolo de siguiente nivel: TCP o UDP.

2.2.3. Encabezado principal de IPv6

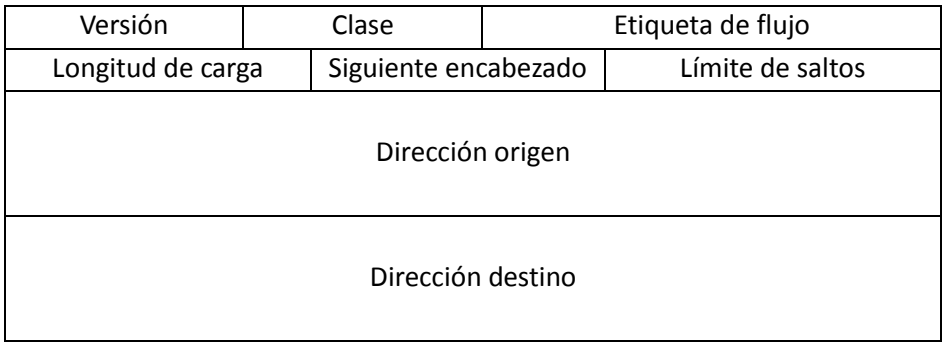


Figura 2.3 Formato del encabezado IPv6

El encabezado principal de IPv6 consta de 40 octetos y se encuentra definido por los siguientes campos:

- Versión (4 bits). La versión IP, este campo contiene el valor 6, que identifica a IPv6.
- Clase (8 bits). Campo con funciones similares al campo Tipo de Servicio en IPv4. Este campo etiqueta un paquete IPv6 con un “Differentiated Services

Code Point" (DSCP, RFC 2474 [7]), que especifica la forma en que el paquete debe ser manejado.

- Etiqueta de Flujo (20 bits). En este campo se etiqueta un flujo de paquetes IPv6, es decir, identifica los paquetes que requieren un mismo trato para facilitar el soporte en tiempo real. Un nodo emisor puede etiquetar secuencias de paquetes con un conjunto de opciones. Los ruteadores guardan el registro de los flujos y pueden procesar paquetes que pertenecen al mismo flujo de manera más eficiente porque no tienen que volver a procesar cada encabezado de los paquetes.
- Longitud de carga (16 bits). Este campo representa la longitud de carga del paquete, la cual consta de todo lo que continúa después del encabezado principal de IPv6.
- Siguiente encabezado (8bits). Sirve para identificar al encabezado que sigue inmediatamente después del encabezado principal de IPv6.
- Límite de Saltos (8 bits) Este campo es análogo al tiempo de vida (time to live, TTL) en IPv4, pero a diferencia de aquel expresa el número de saltos que un paquete puede permanecer en la red antes de ser descartado. En cada uno de los nodos que reenvíe el paquete se decrementa este número en uno para llegar a una cuenta de cero y entonces poder eliminar el paquete.
- Dirección Origen (128 bits). Este campo contiene la dirección IPv6 del nodo que originó el paquete.
- Dirección Destino (128 bits). Este campo contiene la dirección IPv6 del nodo que se espera sea el destino final del paquete.

2.2.4. Encabezados de extensión IPv6

En IPv6, la información de los campos opcionales de Internet es codificada en encabezados separados, éstos pueden ser colocados entre el encabezado principal de IPv6 y el encabezado de las capas superiores del paquete. Esto da un pequeño número de encabezados de extensión, cada uno identificado por un valor distinto.

Los encabezados de extensión ofrecen información extra hacia el destino o sistemas intermedios, como se muestra en la figura 2.4, a lo largo de la ruta cuando así se requiera, para así promover mejoras importantes en el procesamiento.

Encabezado IPv6 Siguiente encabezado = TCP	Encabezado TCP +datos

Encabezado IPv6 Siguiente encabezado = Routing	Encabezado Routing Siguiente encabezado = TCP	Encabezado TCP +datos

Encabezado IPv6 Siguiente encabezado = Routing	Encabezado Routing Siguiente encabezado= fragmentar	Encabezado fragmentar Siguiente encabezado = TCP	Encabezado TCP +datos

Figura 2.4 Encabezados de extensión IPv6.

Con excepción del encabezado de salto a salto, los encabezados de extensión no se examinan o transforman en cualquier nodo a lo largo del camino del paquete, sólo hasta que el paquete alcanza el nodo final (o conjunto de nodos, en caso de multicast). Donde, después de una demultiplexación en el campo del siguiente encabezado, se invoca al primer encabezado de extensión, o el encabezado de la capa superior, si no hay presente un encabezado de extensión.
Cuando un paquete IPv6 utiliza múltiples encabezados de extensión, éstos deben seguir el siguiente orden:

1. Encabezado básico o principal IPv6.
2. Encabezado de opciones de salto a salto.
3. Primer Encabezado de Opciones de Destino.
4. Encabezado de Enrutamiento.
5. Encabezado de Fragmento.
6. Encabezado de Autenticación.
7. Encabezado de Encapsulamiento de Seguridad de la Carga Útil.
8. Segundo Encabezado de Opciones de Destino.
9. Encabezado de Capas superiores (TCP, UDP…).

2.2.5. Impacto de IPv6 en otras capas de TCP/IP.

El desarrollo de IPv6 por los grupos de trabajo de la IETF, estuvo pensado para que la transición de la vieja versión del protocolo, pudiera darse de la forma más sencilla posible y con ello convivir con los diferentes modelos de red como el OSI y el TCP/IP, por lo que en el modelo TCP/IP los cambios más significativos se llevan a cabo en la capa de red, ya que el cambio está dado directamente en el direccionamiento de los hosts.

Las capas superiores del modelo TCP/IP no sufrieron cambios significativos, ya que pueden convivir con IPv4, como con IPv6, así se puede mencionar que las capas de transporte y sesión, pueden llevar a cabo su función sin ningún problema, pues los dispositivos que trabajan en esos niveles pueden interpretar los enlaces entre las capas y tener un manejo óptimo de las direcciones, en las redes se encuentran diferentes implementaciones, que pueden manejar sin problema direcciones que varían en tamaño.

En la capa superior como en la de aplicación, el cambio se ha venido dando de forma gradual, ya que para los usuarios un cambio a la nueva versión más reciente del protocolo se debe de dar de forma transparente, debido a que los usuarios finales no deben de preocuparse por realizar algún cambio en sus aplicaciones para trabajar normalmente en Internet o con aplicaciones que se comuniquen por alguna red. Así que este trabajo se debe llevar acabo por los diferentes desarrolladores de las aplicaciones, así como las óptimas configuraciones de los servicios por parte de los administradores, ya que con esto se tendrá una base sólida para la transición a la versión 6 del protocolo, debido a que en nuestros días muchas de las aplicaciones como navegadores Web, FTP, Telnet, SSH, Servidores de correo, Servidores Web, reproductores de video entre otras, ya cuentan con soporte IPv6 y realizan de forma inherente el mapeo de direcciones, para trabajar con IPv6 sin ningún problema. En la actualidad, las aplicaciones que más se utilizan, cuentan con soporte IPv6, pero todavía queda un largo camino para que todas las aplicaciones cuenten con este soporte, debido a varias razones, como por ejemplo, el API (Application Programmer Interfaces) no se le ha hecho en la mayoría de las veces el cambio para poder interactuar con redes IPv6, y que otras aplicaciones usan restricciones de acceso basadas en la dirección IP.

2.3. Direccionamiento IPv6.

2.3.1. Introducción.

En esta sección se abarcan detalles acerca del direccionamiento en IPv6, ya que en la sección anterior, se presentaron conceptos básicos acerca del protocolo, estos conceptos se aplicarán a continuación para presentar más características que hacen de IPv6 una opción óptima para el futuro de las redes.

El esquema de direccionamiento se define en el "IPv6 Addressing Architecture specification" de la IETF especificado en el RFC 4291 [8], en el cual se define el alcance de las direcciones, dónde éstas pueden ser usadas, las diferentes implementaciones de IPv6, tipos de direcciones, etc. Dos ventajas de IPv6 que se pueden mencionar como resultado de este RFC, son el soporte de multicast de forma intrínseca, que fue requerido en las especificaciones del protocolo, y que los nodos pueden crear direcciones de enlace local durante la inicialización.

2.3.2. Tipos de direcciones IPv6.

Para IPv4 se conocen las direcciones unicast, broadcast y multicast. Para el caso de IPv6 las direcciones broadcast ya no son utilizadas, esto es un punto a favor de las direcciones IPv6, ya que el broadcast es un problema en muchas redes actualmente. Las direcciones anycast, son un tipo más reciente de direcciones que se definieron en el RFC 1546 [9].

2.3.2.1 Direcciones Unicast.

Una dirección unicast es la que identifica a una sola interfaz, la cual envía un paquete a otra única interfaz (uno a uno), esto asemeja a las direcciones IPv4 actuales. Existen varios tipos de direcciones unicast de las cuales se puede mencionar las siguientes:

a) Direcciones locales. Este tipo de direcciones sirven para identificar una interfaz dentro de un mismo segmento de red de área local LAN, por sus siglas en inglés (Local Area Network) o dentro de un nodo, fuera de ellos pierden totalmente su valor. Otro propósito de éstas direcciones, es que se utilizan para auto-configuración (mediante el identificador de interfaz), descubrimiento de vecino o situaciones en las que no hay ruteadores. Por tanto, se ratifica que no pueden retransmitir ningún paquete con dirección fuente o destino que sean de enlace local. El formato de las direcciones de enlace local puede verse en la figura 2.5, que se conforma con el prefijo FE80::/10, del bit 11 al 64 son puestos a cero, 54 bits y finalmente se añade el identificador de interfaz en el formato de Identificador Extendido Único (EUI-64) en los últimos 64 bits de la dirección.

10 bits	54 bits	64 bits
1111111010	0	Identificador de interfaz

Figura 2.5 Formato de las direcciones locales.

b) Direcciones Globales. Las direcciones Unicast Globales son direcciones de Internet, es decir, tienen significado y pueden ser enrutadas en el Internet, ya sea de manera nativa si así lo permite la infraestructura de red, o por medio de túneles. Estas direcciones son agregables con máscaras de bits contiguos, similares al caso de IPv4, Enrutamiento Inter-Dominios sin Clases, CIDR por sus siglas en inglés (Classless Interdomain Routing). Los nodos IPv6 pueden no tener ningún conocimiento o mínimo de la estructura interna de las direcciones IPv6, dependiendo de su misión en la red (por ejemplo, host frente a ruteador). El prefijo de dirección que se utiliza actualmente para direcciones globales es 2000::/3 y la estructura de una dirección global se muestra en la figura 2.6 y se compone por:

- Los primeros n bits de la dirección son el prefijo de enrutamiento que especifica la ubicación de nuestra red (Los primeros tres bits de este prefijo deben ser 001 en notación binaria). Estos n bits representan el fragmento de topología pública de la dirección, el cual representa a los pequeños y grandes proveedores de servicio de Internet en IPv6 y que es controlado por estos ISPs a través de la asignación de la Agencia de Asignación de Números de Internet, IANA por sus siglas en inglés (Internet Assigned Numbers Authority).
- Los siguientes m bits son el identificador de la subred. En las redes actuales se utiliza este fragmento para especificar hasta 65, 535 subredes distintas para enrutamiento dentro del sitio de la organización. Estos m bits representan el fragmento de la topología del sitio de la dirección, el cual está bajo el control de la red.
- Los bits finales son el identificador de la interfaz y especifican una interfaz única dentro de cada subred.

3 bits	n bits	m bits	125-(n+m) bits
001	Prefijo de enrutamiento	Identificador de subred	Identificador de interfaz

Figura 2.6 Formato de las direcciones globales.

Dispositivos más sofisticados pueden tener un conocimiento más amplio de la red, sus límites, etc., dependiendo de la posición misma que un host o ruteador, ocupe en la propia red.

El identificador de interfaz se emplea, por tanto, para identificar interfases, debe de ser único en dicho enlace y también en el ámbito más amplio. En muchos casos, el identificador de interfaz coincidirá con la dirección de la capa de enlace de dicha interfaz. El mismo identificador de interfaz puede ser empleado en múltiples interfases del mismo nodo, sin afectar a su exclusividad global en el ámbito IPv6, siempre y cuando se use un identificador de red diferente en cada caso.

2.3.2.2 Direcciones Anycast.

Una dirección anycast identifica múltiples interfases, pero ésta no hará un broadcast a todo el conjunto, ya que cuando un paquete es enviado a una dirección anycast se entrega a una de las interfases identificadas por dicha dirección, la más cercana de acuerdo a la métrica del protocolo de enrutamiento. Si la dirección Multicast define una comunicación "uno a muchos", una dirección anycast se define como "uno a uno-entre-muchos". Para que los paquetes se entreguen a la dirección anycast más cercana, el ruteador de la red debe conocer qué interfaz tiene asignada una dirección anycast y sus distancias.

Las direcciones anycast, no tienen un espacio propio dentro del direccionamiento IPv6, utilizan el mismo espacio que las direcciones unicast, lo que muestra, es que no se puede diferenciar entre dirección anycast y unicast, con esto el ámbito de las direcciones anycast se equipara con el de unicast, de tal forma que pueden existir direcciones anycast de ámbito de sitio, de enlace o globales. Y una de las características a remarcar sobre estas direcciones es que sólo pueden usarse como direcciones de destino, jamás como origen.

Existe una dirección anycast, requerida para cada subred, que se denomina "dirección anycast del ruteador de la subred" (subnet-router anycast address), su sintaxis es equivalente al prefijo que especifica el enlace correspondiente de la dirección unicast, siendo el indicador de la interfaz igual a cero. Así la dirección Anycast del ruteador se puede representar como en la figura 2.7.

n bits	128-n bits
Prefijo de la subred	0

Figura 2.7 Formato de la dirección anycast del ruteador de la subred.

Los paquetes enviados a la "dirección anycast del ruteador de la subred", serán enviados a un ruteador de la subred, por lo que todos estos deben soportar este tipo de direcciones.

El motivo de ser de estas direcciones es para implementar los siguientes mecanismos:

- *Comunicación con el servidor más cercano*. Estas direcciones permiten que un cliente pueda comunicarse con un servidor de entre un grupo, y que la red le seleccione el más cercano.
- *Descubrimiento de servicios*. Al configurar un nodo con IPv6, no haría falta especificarle la dirección del servidor DNS, Proxy, etc., podría existir una dirección Anycast que identificara a esos servicios.
- *Movilidad*. Nodos que tienen que comunicarse con un ruteador, del conjunto disponible en su red.

2.3.2.3 Direcciones Multicast.

Una dirección Multicast identifica a un conjunto de interfases, estas direcciones están descritas en la RFC 2375 [10]. Tienen un prefijo 1111 1111, después tienen un campo de bandera de 4 bits, de los cuales los tres primeros están reservados y deben ser inicializados a 0, el último bit puede estar en 0, lo cual indica una dirección *multicast* asignada permanentemente, o en 1, si es una dirección *multicast* asignada transitoriamente. El campo que le sigue al de bandera es también de 4 bits y se denomina alcance o ámbito; Su valor se utiliza para limitar el ámbito del grupo de *multicast* (global, local de nodo, local de enlace, local de sitio, etc.). Finalmente, el campo de grupo de 112 bits, identifica el grupo de *multicast,* como se muestra en la figura 2.8.

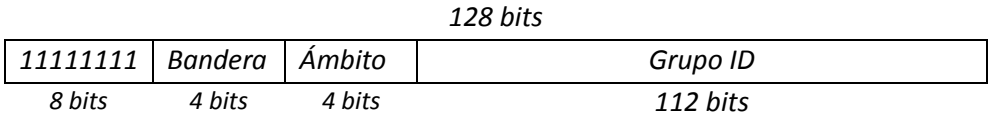


Figura 2.8 Formato de la dirección multicast.

El campo de ámbito como se había mencionado tiene un tamaño de 4 bits que es usado para limitar el alcance del grupo multicast. Los valores del campo ámbito se muestran en la tabla 2.5.

Valor	Significado
0	Reservado
1	Alcance de nodo local
2	Alcance de enlace local
3	Sin asignar
4	Sin asignar
5	Alcance del sitio local
6	Sin asignar
7	Sin asignar
8	Alcance de organización local
9	Sin asignar
A	Sin asignar
B	Sin asignar
C	Sin asignar
D	Sin asignar
E	Alcance global
F	Reservado

Tabla 2.5. Valores del campo ámbito de la dirección Multicast.

El campo “Grupo ID” identifica el grupo multicast, que sea permanente o transitorio, dentro del alcance dado. Las direcciones multicast no pueden ser usadas como direcciones fuente en los datagramas IPv6 ó aparecer en ningún encabezado de ruteo. El RFC 2373 [11] documenta por primera vez las direcciones multicast reservadas o predefinidas y actualizadas en el RFC 4291 [8]. En la tabla 2.6 se presentan las direcciones Multicast reservadas.

FF00:0:0:0:0:0:0:0	Reservado
FF01:0:0:0:0:0:0:0	Reservado
FF02:0:0:0:0:0:0:0	Reservado
FF03:0:0:0:0:0:0:0	Reservado
FF04:0:0:0:0:0:0:0	Reservado
FF05:0:0:0:0:0:0:0	Reservado
FF06:0:0:0:0:0:0:0	Reservado
FF07:0:0:0:0:0:0:0	Reservado
FF08:0:0:0:0:0:0:0	Reservado
FF09:0:0:0:0:0:0:0	Reservado
FF0A:0:0:0:0:0:0:0	Reservado
FF0B:0:0:0:0:0:0:0	Reservado
FF0C:0:0:0:0:0:0:0	Reservado
FF0D:0:0:0:0:0:0:0	Reservado
FF0E:0:0:0:0:0:0:0	Reservado
FF0F:0:0:0:0:0:0:0	Reservado

Tabla 2.6 Direcciones Multicast reservadas.

Las direcciones multicast que se comentaron en esta sección, están reservadas y nunca deben ser asignadas a ningún grupo multicast.También existen grupos definidos para multicast y estos son:

Todas las direcciones de los nodos:

FF01::1

FF02::1

Las direcciones multicast antes mencionadas, identifican el grupo de todos los nodos IPv6 dentro del alcance 1 (nodo local) ó 2 (enlace local). Por ejemplo, las direcciones FF02::1 puede llamar a todo los nodos en este enlace.

Todas las direcciones de los ruteadores:

FF01::2

FF02::2

FF05::2

Las direcciones multicast antes mencionadas, identifican el grupo de todos los ruteadores IPv6 dentro del alcance 1 (nodo local), 2 (enlace local) ó 5 (sitio local).

Por ejemplo la dirección FF02::2 tiene el significado todos los ruteadores en un enlace.

Direcciones de nodo solicitado:
FF02::FFXX:XXXX

Esta dirección multicast es compuesta como una función de direcciones unicast y anycast, está formada tomando los 24 bits de orden inferior de la dirección (unicast o anycast) y agregando esos bits al prefijo de 104 bits FF02:0:0:0:0:1:FF00::/104.
Esto resulta en una dirección multicast en el rango:
FF02:0:0:0:0:1:FF00:0000 hasta FF02:0:0:0:0:1:FFFF:FFFF.
Por ejemplo, la dirección multicast de nodo solicitado que corresponde a la dirección IPv6 4037::01:800:200E:8C6C es FF02::1:FF0E:8C6C.
Las direcciones IPv6 que sólo difieren en los bits de nivel superior, por ejemplo, debido a múltiples prefijos de nivel superior asociados con diferentes agregaciones (proveedores), se mapearán a la misma dirección de nodo solicitado. Esto reduce el número de direcciones multicast que un nodo debe conectar o juntar. Un nodo es requerido para calcular y soportar a una dirección multicast del nodo solicitado por cada dirección unicast y anycast que son asignadas.

2.4. Mecanismos de transición de IPv4 a IPv6

En la actualidad la infraestructura de las redes no está preparada para la migración a IPv6, ya que este proceso se estima tardará un largo tiempo en realizarse, en la mayor parte de las veces es por eso que se han ideado diferentes mecanismos de transición, para que IPv6 pueda coexistir y comunicarse a través de redes IPv4 y con nodos que sólo tengan este soporte. A continuación se describirán los principales mecanismos de transición que impulsan la transición y ayudan a la comunicación usando ambas versiones, los cuales son:

- *Pila dual IPv4/IPv6.*
- *Túneles.*
- *Traducción.*

2.4.1. Pila dual IPv4/IPv6.

En el esquema de la pila dual IPv4/IPv6, que se encuentra definido en el RFC 4213 [12], un nodo de red incorpora ambas versiones, IPv4 e IPv6 que se encuentra en una pila dual Figura 2.9. Una aplicación IPv4 utiliza la pila IPv4 mientras que una aplicación IPv6 utiliza la pila IPv6, así el flujo de decisión en el nodo se basa en el campo de versión del encabezado IP; Para los paquetes que se reciben de las capas inferiores, cuando se evalúa el campo de versión y éste tiene un valor de cuatro pasa la unidad de datos de IP a la pila de IPv4 y si el valor es 6 va a la pila IPv6. Cuando se envían paquetes del tipo

de dirección destino recibido de las capas superiores, ésta es la que determina cuál de las pilas es la más adecuada. Los tipos de direcciones vienen típicamente de la búsqueda del DNS, la pila se escoge apropiadamente, debido al tipo de registro enviado por el DNS.

Muchos sistemas operativos ya proporcionan la pila dual de IP, por ejemplo Microsoft Windows XP, Windows server 2003 y posteriores, son sistemas operativos que implementan la arquitectura de pila dual. En consecuencia, el mecanismo de pila dual puede ser implementado como un mecanismo de transición, sin embargo, hay que tener en cuenta que la pila dual sólo sirve como una aplicación de comunicación, por ejemplo de IPv6 a IPv6 y de IPv4 a IPv4.

<table>
<tr><td>Aplicación IPv4</td><td>Aplicación IPv6</td></tr>
<tr><td colspan="2">Socket API</td></tr>
<tr><td>TCP/UDP IPv4</td><td>TCP/UDP IPv6</td></tr>
<tr><td>IPv4</td><td>IPv6</td></tr>
<tr><td colspan="2">Capa 2</td></tr>
<tr><td colspan="2">Capa 1</td></tr>
</table>

Figura 2.9 Mecanismo de transición pila dual o doble**.**

2.4.2. Túneles.

El mecanismo de túneles es usado para desplegar el envío de paquetes IPv6, si la infraestructura de una red es completamente hecha para IPv4. Los túneles pueden ser usados para llevar tráfico IPv6 encapsulado en paquetes IPv4 y en este caso el túnel es sobre una infraestructura IPv4. En primera instancia si un proveedor de Internet sólo tiene una infraestructura IPv4, el túnel permite tener un comportamiento de una red IPv6, por lo que el túnel implementado a través de las redes IPv4 de los ISP's alcanza otro host o red IPv6.

Una de las principales desventajas es el retardo adicional ocasionado por el encapsulado y desencapsulado de paquetes IPv6 en datagramas IPv4, así como el tráfico de un mayor número de paquetes ocasionado por la reducción de espacio para datos en los datagramas IPv4 que contienen dentro paquetes IPv6.

El mecanismo de túnel y encapsulamiento se encuentran definidos en los RFC 2473 [13], 4213 [12] y 3056 [15] con dos diferentes tipos de túneles:

- Manuales.
- Automáticos.

Por lo que el proceso de encapsulamiento en un túnel se ve ejemplificado en la figura 2.10.

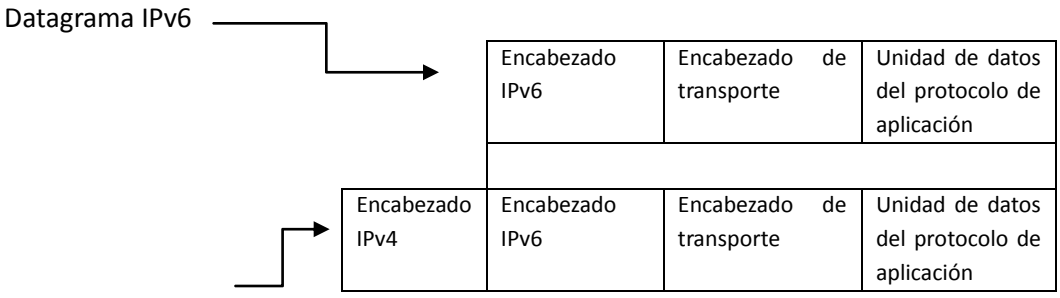


Figura 2.10. IPv6 encapsulado en IPv4.

2.4.3. Traducción.

Este mecanismo de transición es usado para nodos que sólo cuentan con la pila IPv6 habilitada, y se encuentran dentro de una red IPv6, y desean comunicarse con otro nodo que solo tiene la pila IPv4 habilitada dentro de una red IPv4. Sin embargo, esta técnica requiere tener habilitados mecanismos de traducción entre IPv4 e IPv6 en las orillas de ambas redes (ruteadores). La principal desventaja es que todo el peso de esté mecanismo de transición recae en los dispositivos encargados de hacer dicha traducción, a los que no siempre se tiene acceso.

2.5. Conexiones automáticas y manuales.

En el proceso de transición a IPv6, se puede decir que todavía queda un largo camino por recorrer para sustituir totalmente a IPv4, ya que muchas de las redes en la actualidad no soportan la nueva versión protocolo, no se tiene un conocimiento completo sobre los conceptos ni la capacitación para realizar configuraciones y conexiones utilizando IPv6, por lo que los administradores de las redes deben tener en cuenta el panorama que se tiene sobre los desarrollos de la transición de IPv4 a IPv6; tener en cuenta la demanda de direcciones por parte de los usuarios, que las nuevas tecnologías necesitarán un cambio a la versión seis del protocolo y deberán prepararse para adoptarlo. Es por ello que cuando se toma la decisión de entrar de lleno a la utilización de IPv6, se puede considerar que IPv6 es "Plug & Play", ya que cada host mediante una serie de pasos automáticos puede autoconfigurarse, con esta característica se puede facilitar el uso de la nueva versión IP, para así poder hacer más transparente la transición en los equipos de los usuarios finales y con poca dificultad a los administradores de red.

En una conexión utilizando direcciones IPv6 se debe pensar en cómo se obtendrán las direcciones, ya que es de suma importancia que éstas no se dupliquen y puedan utilizarse para obtener conectividad con otros host en un segmento. Las direcciones pueden obtenerse de forma automática por medio de DHCPv6 que lo realiza el método "Con Estado" (Statefull), o con el método "Sin Estado" (Stateless).

En la definición de IPv6, se establece el proceso por el cual se pueden generar las direcciones de enlace local y direcciones globales, mediante el método Sin estado, con el cual también se pueden detectar las direcciones duplicadas.

El método *Sin estado* no requiere de ninguna configuración en el host, ya que el proceso lo lleva a cabo el ruteador, la configuración de éste es mínima o nula, y no requiere servidores adicionales. También permite al host generar su propia dirección mediante una combinación de información disponible localmente e información proporcionada por el ruteador. A grandes rasgos el proceso que lleva acabo el ruteador es el de anunciar los prefijos que identifican la subred, subredes asociadas o conectadas a sus interfases, mientras que el host genera un "identificador de interfaz", que identifica de forma única a al equipo en la subred. Y es con estos componentes mediante los cuales se puede formar la dirección. Si en la subred donde se encuentre un host, carece de un ruteador, sólo podrá generar la dirección de enlace local, aunque con esta dirección será suficiente para poder comunicarse entre los mismos hosts conectados en la subred.

Por otra parte la configuración "Con estado", es donde el host obtiene la dirección de la interfaz y/o la información y parámetros de configuración desde un servidor. Los servidores mantienen una base de datos con las direcciones que han sido asignadas a cada host.
Ambos métodos por los cuales se configuran los hosts se complementan, ya que un host puede utilizar autoconfiguración *Sin estado* para generar su propia dirección y obtener el resto de los parámetros mediante la configuración *Con estado*.

La autoconfiguración está pensada sólo para asignar direcciones a los hosts en una subred, no a los ruteadores, aunque esto no implica que parte de la configuración de los ruteadores también pueda ser realizada automáticamente, que por lo general es la dirección de enlace local.

2.5.1. La configuración Sin estado

Es el procedimiento que se diseñó en base a los siguientes preceptos:

- Evitar la configuración manual de dispositivos antes de su conexión a la red. Con lo cual se requiere, un mecanismo que permita a los hosts obtener o crear direcciones únicas para cada una de sus interfases, asumiendo que cada una puede proporcionar un identificador único para si misma, este puede ser combinado con un prefijo para formar una dirección válida e distintos ámbitos.
- Las pequeñas redes o dominios, con máquinas conectadas a un único enlace, no deberían requerir la presencia de un servidor "Con estado" o un ruteador, como requisito para comunicarse entre ellas, ya que para obtener características de "plug & play", se emplean direcciones de enlace local, dado que tienen un prefijo perfectamente conocido que identifica el único enlace compartido, al que se conectan todos los hosts. Cada dispositivo forma así una dirección de enlace local anteponiendo el prefijo de enlace local a su identificador de interfaz.
- Las redes o dominios grandes, con múltiples subredes y ruteadores, tampoco requieren la presencia de un servidor "Con estado", como requisito para comunicarse, para esto los hosts han de determinar los parámetros, para generar sus direcciones globales o de enlace local y los prefijos que identifican las subredes a las que se conectan. Los ruteadores generan mensajes periódicos que anuncian a estos prefijos, que incluyen opciones como listas de prefijos activos en los enlaces.

- La configuración de direcciones debe de facilitar la renumeración de los dispositivos de un sitio.
- Sólo es posible utilizar este mecanismo en enlaces capaces de funcionar con multicast, y comienza por tanto cuando es iniciada o activada una interfaz que permite multicast.
- Los administradores de sistemas necesitan la habilidad de especificar qué mecanismo de autoconfiguración debe ser usado. Los mensajes de anuncio de los ruteadores incluyen indicadores para esta función.

Los pasos básicos para la autoconfiguración de una interfaz cuando ha sido conectada a un segmento de red son:

- Se genera una posible dirección de enlace local, como se ha mencionado con anterioridad.
- Se verifica que dicha dirección "posible" puede ser asignada, al no encontrarse duplicada en el enlace.
- Si se encuentra duplicada, la autoconfiguración se detiene, y se requiere un procedimiento manual.
- Si no se encuentra duplicada, la conectividad entre las direcciones se puede lograr, al asignarse definitivamente dicha dirección "posible" a la interfaz en cuestión.
- Si se trata de un host, se cuestiona a los posibles ruteadores para indicarle al host lo que debe de hacer.
- Si no hay ruteadores, se llama al procedimiento de autoconfiguración con estado.
- Si hay ruteadores, éstos contestarán indicando fundamentalmente, cómo obtener las direcciones si se ha de utilizar el mecanismo "Sin estado" u otra información.

2.5.2. La configuración Con estado

Se implementa por medio de un servidor de DHCP para IPv6 que usa el protocolo UDP en un esquema cliente/servidor, diseñado para reducir el costo de gestión de nodos IPv6 en entornos donde los administradores precisan un control en la asignación de los recursos de la red, superior al implementado por el mecanismo de configuración sin estado.

Los mecanismos de configuración pueden usarse de forma concurrente y así reducir el costo de propiedad y administración de la red, por lo que para lograr esta característica, se centraliza la gestión de los recursos de la red, tales como direcciones IP, información de ruteo, información de instalación de sistemas operativos, etc.; sobre uno o varios servidores DHCP, en lugar de distribuir dicha información en ficheros de configuración locales en cada nodo.

Por lo que DHCP ha sido diseñado para ser fácilmente extensible con nuevos parámetros de configuración, a través de extensiones que incorporan esta nueva información. Los objetivos de DHCPv6 son:

- DHCP es un mecanismo, no una política. La política es establecida por el administrador de la red y DHCP le permite propagar los parámetros adecuados, según dicha política.
- DHCP es compatible, lógicamente, con el mecanismo de configuración "sin estado".
- DHCP no requiere configuración manual de parámetros de red en clientes DHCP, excepto en casos donde dicha configuración se requiere debido a medidas de seguridad.
- DHCP no requiere un servidor en cada enlace, dado que debe funcionar a través de redes DHCP.
- DHCP coexiste con nodos configurados estáticamente, así como con implementaciones existentes en la red.
- Los clientes DHCP pueden operar en enlaces donde no hay ruteadores IPv6.
- Los clientes DHCP proporcionan la habilidad de renumerar la red.
- Un cliente DHCP puede hacer múltiples y diferentes peticiones de parámetros de configuración, de uno o varios servidores DHCP simultáneamente. DHCP proporciona suficiente información para permitir a los servidores DHCP el seguimiento del estado de configuración de los clientes.
- DHCP incorpora los mecanismos apropiados de control de tiempo y retransmisiones para operar eficazmente en entornos con una alta latencia y/o reducido ancho de banda.

Los cambios fundamentales entre DHCPv4 y DHCPv6 están basados en el soporte inherente del formato de direccionamiento y autoconfiguración IPv6; son las siguientes:

- La dirección de enlace local permite a un nodo tener una dirección tan pronto como arranca, lo que significa que todos los clientes tienen una dirección IP fuente para localizar un servidor o relay en su mismo enlace.
- Los indicadores de compatibilidad BOOTP y broadcast han desaparecido.
- El multicast y los ámbitos de direccionamiento permiten el diseño de paquetes de descubrimiento, que definen por si mismos su rango por la dirección multicast, para la función requerida.
- La autoconfiguración statefull ha de coexistir e integrarse con stateless, soportando la detección de direcciones duplicadas y los tiempos de vida de IPv6, para facilitar la renumeración automática de direcciones y su gestión.
- Se soportan múltiples direcciones por cada interfaz.

Capítulo 3
Mecanismo de transición por túneles

3.1. Introducción.

Los túneles son de gran ayuda en el esquema de transición de IPv4 a IPv6, su función principal es llevar protocolos incompatibles o datos en particular sobre una red no apta para ellos, y así crear redes privadas virtuales. Dado su función, se han utilizado como mecanismo de transición para comunicar islas IPv6 en la Internet actual, que estaba basada casi en su totalidad en IPv4.

Los túneles IPv6 en IPv4 usualmente están hechos por la adición de un encabezado IPv4 antes de un paquete Ipv6. El paquete resultante es posteriormente enviado a la dirección destino que se encuentra en el encabezado IPv4. Al llegar a su destino la cabecera es retirada y el paquete se procesa como si se hubiera recibido un paquete IPv6 común.

3.2. Túneles manuales.

En los túneles configurados, la dirección del extremo del túnel es determinada por la información de la configuración en el nodo encapsulado. Este túnel está definido en el RFC 4213 [12]. Para cada túnel, el nodo encapsulado debe guardar la dirección de extremo del túnel. Cuando un paquete IPv6 es transmitido por un túnel, la configuración de la dirección del extremo para ese túnel es usada para ser la dirección destino por el encabezado IPv4.
La determinación de cómo los paquetes irán por el túnel está hecha por la información de ruteo en el nodo de encapsulación. Esto es usualmente hecho con una tabla de ruteo, que dirige los paquetes basados en la dirección destino usando un prefijo y técnicas de encuentro.
Los hosts IPv6/IPv4 que se encuentren conectados a un segmento de red con ruteadores que no soportan IPv6, se les puede configurar un túnel, para que así puedan alcanzar un ruteador IPv6. Este túnel permite a los host comunicarse con el resto de las redes IPv6.
Si se conoce la dirección IPv4 de un ruteador de pila dual que se encuentra en el backbone IPv6, ésta puede ser usada como la dirección de extremo del túnel. El túnel puede ser configurado dentro de la tabla de ruteo como un "ruteador IPv6 por defecto". Es decir que todas las direcciones IPv6 pueden pasar a través del túnel.

3.3. Túneles automáticos.

Los túneles automáticos permiten que nodos IPv4/IPv6 puedan comunicarse sobre una red IPv4 sin la necesidad de una preconfiguración de un túnel destino. La dirección de extremo de los túneles está determina por una dirección IPv4 compatible, este tipo de direcciones son asignadas exclusivamente a los nodos que usan un túnel automático.
Los túneles automáticos se definen en el RFC 4213 [12], para este mecanismo el prefijo ::/96 es reservado para las direcciones IPv4 compatibles para la parte más a la derecha, los últimos 32 bits de la direcciones IPv6 son destinados para la dirección IPv4 del nodo. Los paquetes IPv6 son automáticamente encapsulados en un paquete IPv4 y direccionados a una correspondiente dirección IPv4 y destinados a un túnel.

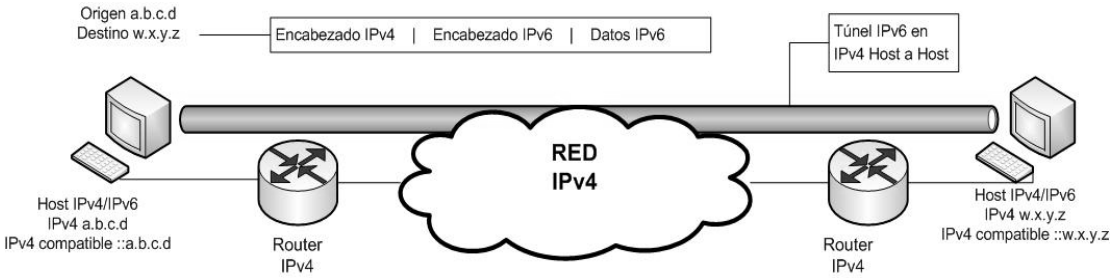


Figura 3.1 Representación de un sistema de túnel.

En la figura 3.1 se puede ver como se da el proceso de encapsulamiento. El host o ruteador de cada extremo del túnel contiene una dirección IPv4 compatible así también soporta IPv4 e IPv6. La tabla de ruteo se fija de tal forma que el prefijo ::/96 es direccionado directamente a la interfaz del túnel automático.

La dirección IPv4 compatible tiene un esquema muy restringido, ya que el enrutamiento no se produce más allá del extremo final del túnel, este punto también es el destinatario, por lo que esta técnica es poco recomendable, ya que se considera obsoleta y es mejor optar por una solución más robusta como es 6to4 ó ISATAP.

3.4. Túnel 6over4 .

El túnel 6over4, también denominado túnel de Multicast IPv4, es una técnica de túnel que se describe en el RFC 2529 [14]. Este túnel esencialmente utiliza el transporte en capa 2 de IPv4 para IPv6. El mecanismo se comporta como una subred Ethernet sobre IPv6 excepto que la funcionalidad de la capa 2 de Ethernet es reemplazada por el Multcast IPv4. El mecanismo no requiere ningún prefijo especial como es el caso de 6to4.

El túnel 6over4 trata a la infraestructura IPv4 como un enlace único con capacidades de multicast, está relación se muestra en las figuras 3.2 y la figura 3.3, donde se puede ver la arquitectura física y la visión lógica para el esquema de transición de 6over4, respectivamente.

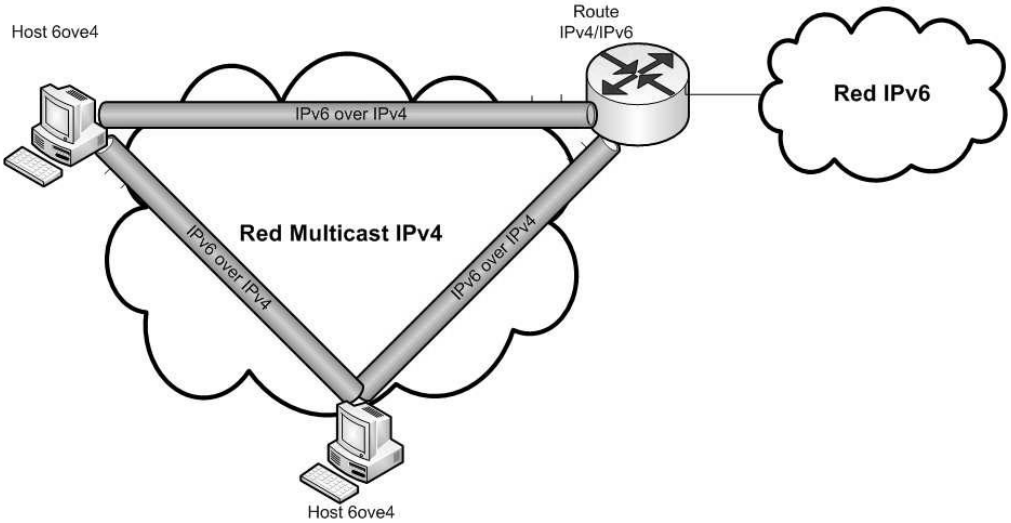


Figura 3.2 Arquitectura física del túnel 6over4.

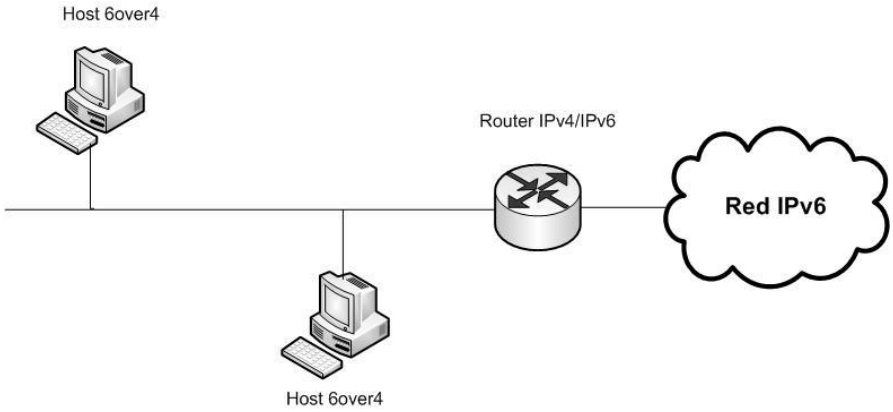


Figura 3.3 Visión lógica del túnel 6over4.

Para una operación más eficiente, un host 6over4 necesita formar un número de direcciones, para la formación de estas direcciones, un nodo 6over4 usa la dirección IPv4 de la interfaz de la misma forma que la interfaz de Ethernet, usa los 64 bit de su identificador único (EUI-64):

- *Dirección Unicast.* Una dirección unicast es formada de la siguiente manera: 6over4 usa un host, un prefijo valido de 64 bits para la dirección unicast y 64 bits del identificador de la interfaz ::[dirección IPv4], donde la dirección IPv4 es de 32 bits que perteneciente al host.
- *Dirección de enlace local.* Por defecto, los hosts 6over4 configuran automáticamente la dirección de enlace local de la forma FE80::[dirección IPv4] en cada interfaz 6over4. Por ejemplo un host con una dirección IPv4 10.0.0.1 tendrá una dirección de enlace local FE80::0A00:0001, o también se puede expresar como FE80::A00:1.
- *Solicitud de una dirección multicast de un nodo.* Para la resolución eficiente de la dirección, al nodo se le asigna una dirección multicast donde se envía un mensaje de solicitud de vecino como parte de la resolución de la dirección, con esto se tiene la ventaja de que no se molestarán a los demás hosts con mensajes para la resolución de la dirección (como suele ocurrir para enlace), ya que esto no va dirigido a ellos y es usado de la siguiente manera:
  En vez de enviar el mensaje de solicitud de vecino en el segmento a cada uno de los nodos (vía un FF02::1 a todos los nodos), el mensaje de solicitud de vecino es enviado a un grupo multicast muy restringido identificado por la dirección multicast del cliente.
- La dirección multicast para una dirección unicast dada, es construida por los últimos tres octetos de la dirección unicast IPv4, teniendo la dirección FF02::1FF00:0000/104 como base para formarla. Así la dirección multicast de un cliente está determinada por su dirección unicast 2001:630:200:8100:02C0:4FFF:FE68:12CB es FF02::1:FF68:12CB y se encuentra ilustrada en la figura 3.4. Esta es la dirección multicast de un cliente, el nodo utiliza como destino un paquete de solicitud de vecino.

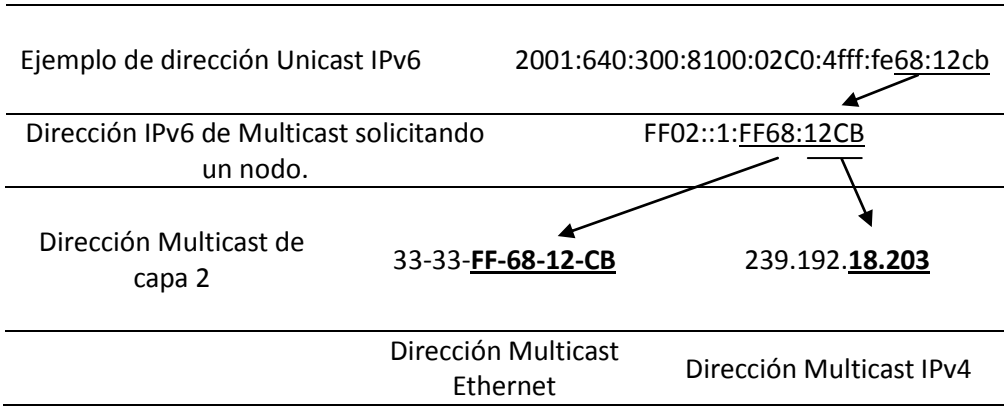


Figura 3.4 Ethernet y esquema Multicast IPv4 para IPv6.

3.5. Túnel 6to4.

El mecanismo de transición 6to4 está hecho para la interconexión entre sitios aislados IPv6 por medio de túneles automáticos en redes IPv4. La motivación para este método es que debe permitir a dominios IPv6 aislados o hosts únicos, que se encuentran conectados a redes IPv4, comunicarse con dominios o hosts IPv6, con la mínima configuración manual.

Los túneles automáticos son logrados por tener un ruteador, llamado ruteador 6to4, en la frontera del dominio IPv6, conectado a la red IPv4. 6to4 opera teniendo la dirección IPv4 de la interfaz del ruteador y una parte del prefijo de la dirección IPv6 que es asignado al host en el dominio IPv6 respectivo.

El sistema 6to4 permite transmitir paquetes IPv6 sobre redes IPv4 sin la necesidad de configurar explícitamente un túnel. Los ruteadores de conversión también tienen un lugar en el mecanismo 6to4, ya que permiten que los hosts puedan comunicarse con otros hosts en redes IPv6.

6to4 no está obligado a ser configurado o que se le de soporte a las redes cercanas al host, cobra una gran relevancia durante las etapas iníciales para la implementación de una conexión IPv6 nativa. Sin embargo, sólo servirá como mecanismo de transición, ya que no está destinado para ser utilizado permanentemente.
El mecanismo puede ser usado por un solo host, o por una red local IPv6. Cuando es usado por un host, el cual tiene una dirección IPv4, éste es responsable de encapsular los paquetes IPv6 de salida y desencapsular los paquetes 6to4 entrantes. Muchos sistemas operativos de diferentes host implementan este encapsulado y desemcapsulado por una pseudo interfaz 6to4.

Cuando 6to4 es usado por una red local, ésta necesita una sola dirección IPv4. Dentro de la red local, los host aprenden las direcciones IPv6 y los ruteadores utilizan el protocolo de descubrimiento de vecino, tal como se hace en una red IPv6 nativa.

El mecanismo 6to4 puede definirse en tres funciones principales:

1. Asignación de un bloque de espacio de direcciones IPv6 a un host o red que tiene una dirección IPv4.
3. Encapsula paquetes IPv6 dentro de paquetes IPv4 y su transmisión sobre redes IPv4 usando 6to4.
4. Determinar rutas de tráfico entre hosts 6to4 y redes IPv6 nativas.

Para cualquier dirección IPv4 de 32 bits de un host puede ser construido un prefijo IPv6 de 6to4 para ser usado. El método de asignación de un prefijo IPv6 a un host en un dominio 6to4 está definido en el RFC 3056 [15], para este propósito la IANA a reservado un espacio específico para las direcciones IPv6 del túnel 6to4, que es el 2002::/16, el resto de la dirección que acompaña al prefijo, se obtiene añadiendo la dirección IPv4 de 32 bits asignada al ruteador externo 6to4. Al realizar este proceso, el esquema de la dirección 6to4 se muestra en la figura 3.5. El prefijo 6to4 puede ser abreviado como 2002:[dirección IPv4]::/48, dentro del dominio IPv6, el prefijo puede ser usado como cualquier otro prefijo valido IPv6.

2002	Dirección IPv4 del ruteador de frontera.	Identificador de subred.	Identificador de interfaz.
16 bits	32 bits	16 bits	64 bits

Figura 3.5 Esquema de la dirección 6to4.

3.6. Túnel Teredo.

Teredo es una tecnología de transición a IPv6, el cual provee de direcciones y túneles host a host con direcciones IPv6 unicast, donde los host trabajan bajo un modelo de pila dual y éstos se encuentran detrás de dispositivos de Traducción de Dirección de Red, NAT por sus siglas en inglés (Network Address Translation). La operación básica de los NAT está definida en el RFC 1631 [16], ya que intenta conservar el direccionamiento IPv4, y así poder envolverlas en un mapeo privado, definir direcciones IPv4 internas, un número de puerto dentro de una subred pública, direcciones externas IP y un número de puerto asignado por el dispositivo.

Para atravesar un NAT IPv4, teredo especifica los paquetes IPv6 enviando un mensaje UDP IPv4-base. Teredo también agrega varias técnicas definidas en el RFC 5394 [17] para el envió por túneles el tráfico UDP por varios tipos de NAT.

Una forma básica de representar la técnica de transición teredo, se muestra en la figura 3.6, ya que un NAT es usado para proveer conectividad a un sitio a través de una sola dirección IPv4 pública. Este tipo de implementaciones se usan en medianas empresas, pequeñas oficinas y otros ambientes.

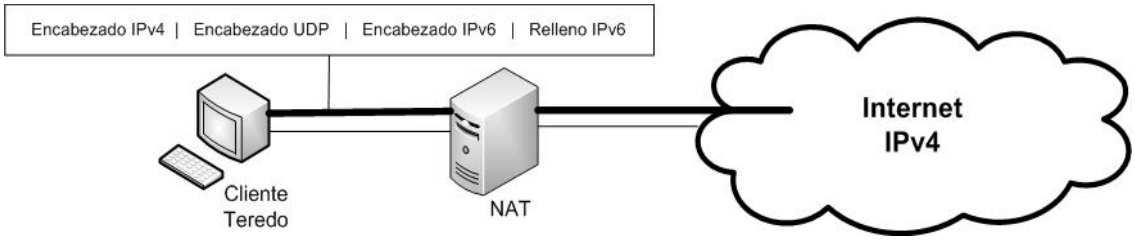


Figura 3.6 Cliente Teredo y túnel.

Una de las diferencias con respecto a 6to4 es que su tecnología de comunicación IPv6 está dada en los ruteadores de frontera mientras que el túnel Teredo se origina en el host y usa IPv4 como tecnología de comunicación.

Los dispositivos NAT causan problemas al mecanismo 6to4, es por eso que la creación de Teredo es de gran importancia, ya que el buen desempeño de 6to4 se basa en la utilización de una dirección IPv4 pública y la implementación de un ruteador 6to4, debido a que en muchas ocasiones las configuraciones de los dispositivos NAT están hechas en varios niveles, por lo que no se podría asignar a cada uno de estos NAT una dirección pública IPv4. Otra razón por la cual Teredo es de gran importancia es debido a que los paquetes IPv6 encapsulados en paquetes IPv4 utilizan el valor 41 en el campo de protocolo en el encabezado del paquete IPv4 y los dispositivos NAT solo son capaces de traducir los protocolos TCP y UDP, debido a que el protocolo 41 no es entendible por los dispositivos NAT, por lo que no puede fluir a través de ellos para llegar a los sitios destino. Teredo hace un buen manejo del protocolo UDP, ya que con éste se auxilia para la creación de túneles, debido a que los dispositivos NAT trabajan bien con este protocolo a múltiples niveles. Con la utilización de una sola dirección IPv4 y del mapeo sobre UDP, se pueden realizar túneles a diferentes hosts con pila dual, detrás de un mismo dispositivo NAT.
Los componentes principales de Teredo se muestran en la figura 3.7 y son:

- *Cliente Teredo:* Son los que se encuentras detrás de un dispositivo NAT los cuales están conectados a redes IPv4 y desean conectarse a redes IPv6, esto lo pueden hacer a través de un servidor Teredo utilizando paquetes UDP.
- *Servidor Teredo:* Es el que está conectado a una red IPv4 y es el que espera la comunicación a las redes IPv6, se encarga de administrar la señal y tráfico de los cliente Teredo, así como asignar las direcciones IPv6 unicast a cada host.
- *Teredo Relay:* Está conectado a redes IPv6 y actúa como ruteador para brindar conectividad a los clientes Teredo.

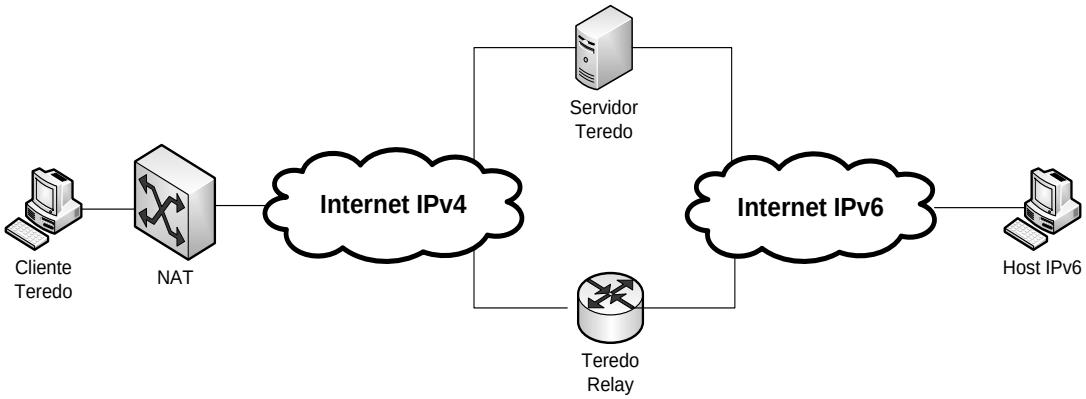


Figura 3.7 Componentes de túnel Teredo.

El formato de la dirección Teredo se muestra en la figura 3.8.

- *El prefijo de Teredo* se le asigna el valor 2001::/32 especificado en el RFC 4380 [18].
- *La dirección IPv4 del servidor teredo* es una dirección pública del servidor y ésta ayuda a la configuración de la dirección IPv6 para el cliente Teredo.
- *El campo de bandera* es resultado del proceso de configuración de la dirección Teredo e indica el tipo de NAT que usa el cliente.
- Los últimos dos campos son el oscuro mapeo externo de las dirección IPv4 y el puerto del cliente Teredo.

Prefijo Teredo	Dirección IPv4 del servidor Teredo	Banderas	Puerto Externo	Dirección Externa
32 bits	32 bits	16 bits	16 bits	32 bits

Figura 3.8 Formato de la dirección Teredo.

3.7. Túnel Broker.

Es un mecanismo establecido por la IETF para facilitar la implementación de túneles configurados en redes IPv4, ya que con esta técnica no se tienen que configurar manualmente los extremos del túnel, como se define en el RFC 3053 [19], en el cual se expresa que este túnel es un sistema que funciona como un servidor sobre la red IPv4 y recibe peticiones de nodos con pila dual para configurar túneles automáticamente. Estas peticiones son enviadas vía HTTP sobre redes IPv4 por el nodo que desea configurar dicho túnel. El túnel broker entonces envía de vuelta al cliente información tal como la dirección IPv4 del servidor del túnel, la dirección IPv6 del servidor del túnel, la nueva dirección IPv6 que será asignada al host con pila dual y las rutas IPv6 por defecto para la configuración del túnel.

La idea del túnel broker es dar un enfoque alternativo para ofrecer servidores dedicados llamados túneles brokers, para así administrar automáticamente túneles que sean solicitados por usuarios, esto es utilizado para que aumenten las expectativas para la interconexión de hosts a través de IPv6 y que permita que los proveedores puedan dar un fácil acceso a sus usuarios a las redes con la nueva versión del protocolo.

La principal diferencia entre el túnel broker y el mecanismo 6to4 es que ocupan un segmento diferente para la comunidad IPv6.

- El túnel broker se adapta a las pequeñas redes IPv6 aisladas y especialmente a los host aislados por redes IPv4, por lo que facilita la conexión entre redes IPv6 existentes.
- El enfoque del túnel 6to4, está diseñado para permitir que sitios aislados IPv6 puedan conectarse fácilmente sin tener que esperar que los proveedores de Internet tengan que ofrecer un servicio nativo de IPv6. Esto es muy adecuado para las extranet y redes virtuales privadas.

El túnel broker puede considerarse como un proveedor de Internet virtual, el cual suministra conectividad IPv6 a los usuarios que se encuentren conectados a la red IPv4, así tener acceso a esta tecnología emergente y conectarse a través de IPv6. Se puede tener acceso a muchos túneles broker que ofrecen este servicio, por lo que los usuarios podrán escoger el más conveniente, tomando en cuenta que la elección será influenciada por la cercanía del servidor de túnel y costo.

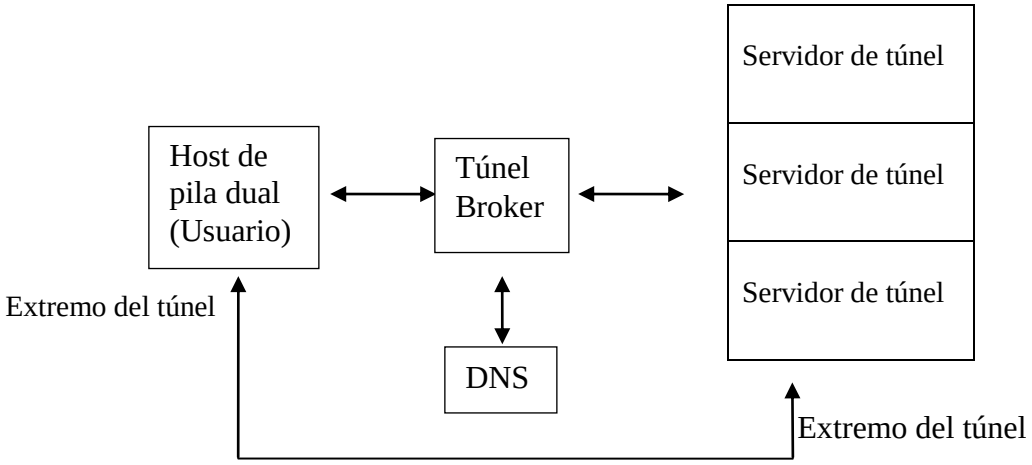


Figura 3.9 Modelo del túnel broker.

En la figura 3.9 se puede ver el modelo de un túnel broker el cual se encargara de gestionar la creación, modificación y supresión de nombres de usuarios, así se puede ver que por razones de escalabilidad el mecanismo puede compartir su carga entre varios servidores, así las ordenes para la configuración son enviadas a los diferentes servidores para realizar dichas acciones. El túnel también puede registrar la dirección y nombre de usuario en un DNS.

El servidor de túnel es un ruteador de pila dual que se encuentra conectado a Internet, una vez que recibe una orden del túnel broker, éste crea, modifica o borra el espacio de cada túnel, así se pueden mantener estadísticas de uso para cada túnel activo.

Los túneles bróker realizan los siguientes puntos:

- Selecciona un servidor de túnel que sirve como punto de salida. Si existe más de una opción, hace la selección en base a criterios preconfigurados.
- Selecciona un prefijo para el cliente. Este prefijo puede ser de cualquier longitud, los más comunes son /48 (prefijo de sitio), /68 (prefijo de subred) y /128 (prefijo de un host)
- Define el tiempo de vida del túnel.
- Registra la dirección IPv6 global en un DNS asignado.
- Configura el servidor de túnel.
- Envía la información de la configuración al cliente, esta información incluye los parámetros del túnel y DNS.

Así se tiene la información necesaria para la configuración del túnel y un cliente puede tener acceso a las redes IPv6 a través del servidor de túnel.

3.8. ISATAP.

El protocolo de direccionamiento de túneles automático intra-sitio, ISATAP por sus siglas en inglés (Intrasite Automatic Tunnel Addressing Protocol), que se encuentra definido en el RFC 4214 [20] es una técnica por la cual se pueden crear túneles IPv6-en-IPv4 de forma automática dentro de un sitio IPv4. Ya que esto lo puede realizar gracias a que cada host solicita al ruteador dentro del sitio IPv4, una dirección IPv6 y diferentes rutas de encaminamiento, con lo cual, los paquetes pueden ser enviados a diferentes sitios IPv6 a través de un ruteador ISATAP, los paquetes que llevan como destino otros host dentro del sitio son entregados directamente a través de túneles ISATAP, este mecanismo es muy similar a 6over4 por las características con las que se entregan los paquetes antes mencionados. Las direcciones IPv6 se configuran automáticamente por el protocolo de descubrimiento de vecino (Neighbor Discovery), aunque éste proceso también puede ser realizado de forma manual. En la figura 3.10 se puede ver la arquitectura de ISATAP.

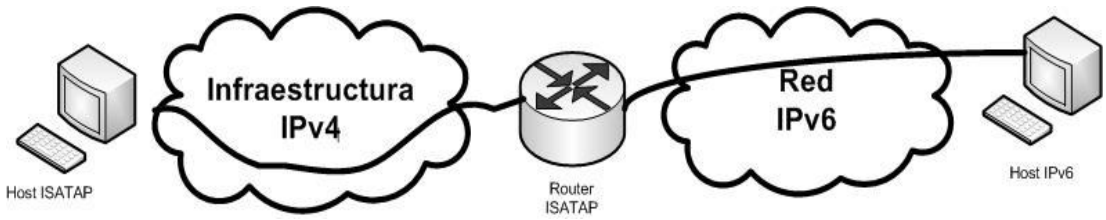


Figura 3.10 Arquitectura ISATAP.

El esquema de la dirección ISATAP se muestra en la figura 3.11, ya que la dirección ISATAP como otros métodos de transición como 6to4 y 6over4 contiene embebida una dirección IPv4, la cual está concatenada con un prefijo global unicast IPv6 y con un identificador de interfaz. El prefijo que utiliza ISATAP para habilitar una dirección de enlace local en un host es FE80::/10. El identificador de la Interfaz es formado por los 32 bits de la dirección IPv4, para después concatenar el identificador especial de ISATAP con el valor 0000:5EFE, que ha sido reservado por IANA para las direcciones ISATAP. Un identificador de interfaz ISATAP es ::05EFE:wx:yz: donde los valores de w.x.y.z son los que corresponden a los valores de la dirección IPv4. Por ejemplo, para una dirección 192.24.12.1 una configuración automática

su dirección de enlace local seria FE80::5EFE:C018:C1, por lo que el valor C018:C1 es la expresión hexadecimal de la dirección IPv4. Las direcciones ISATAP también pueden tener prefijos unicast globales, los cuales son asignados por los ruteadores. En los ruteadores y los hosts ISATAP generalmente se tienen tres tipos de direcciones: unicast IPv6 global, IPv6 enlace local y una dirección IPv4.

Enlace local o asignado por router ISATAP		ID de interfaz.
Prefijo ISATAP	0000:5EFE	Dirección IPv4
64 bits	64 bits	

Figura 3.11 Esquema de la dirección ISATAP.

Capítulo 4
Ruteo en IPv6

4.1. Introducción.

Las redes en la actualidad son muy complejas por lo que pueden catalogarse de muchas formas al contener características diferentes unas de otras, tanto en número de host, topología, tamaño, extensión, etc. Con todo esto se dificulta su identificación, ya que existen Redes de Área Personal, PAN por sus siglas en inglés (Personal Area Network), Redes de Área Local, LAN por sus siglas en inglés (Local Area Network), Redes de Área Metropolitana, MAN por sus siglas en inglés (Metropolitan Area Network), Redes de Área Amplia, WAN por sus siglas en inglés (Wide Area Network) por mencionar algunas, a medida que la tecnología avanza éstas redes son cada vez más complejas y no se podría asegurar su clasificación por área geográfica, como un método de identificación único.

Los ruteadores típicamente se usan para llevar a cabo la comunicación con otros dispositivos pertenecientes a terceros como un ISP y se hace uso del direccionamiento lógico, por lo que trabajan principalmente en capa 3 del modelo OSI. Un ruteador es un dispositivo inteligente, que como cualquier host, es capaz de operar en cualquiera de las siete capas del modelo OSI. La comunicación en las dos primeras capas de este modelo se da principalmente con direccionamiento físico para identificar a los dispositivos que se están comunicando, y así las redes pueden comunicarse directamente con los ruteadores. Pero su característica principal, es que pueden identificar rutas para el encaminamiento de paquetes, basándose en las direcciones de capa tres. Así, trabajan a través de Internet con la interacción de múltiples redes, sin importarles qué tan lejos o cerca se encuentren.

Para que pueda darse la comunicación eficaz entre redes, es necesario por lo menos una ruta física que interconecte las computadoras origen y destino, por lo que éstas computadoras deben tener un lenguaje en común, que es llamado lenguaje enrutado, y los ruteadores que se encuentran en la ruta que comunica a las computadoras también deben tener un lenguaje en común o protocolo de enrutamiento. Este protocolo permite a los ruteadores realizar las siguientes funciones:

- Identificar rutas eficientes para llegar a los hosts y redes específicas.
- Realizar por medio de algoritmos matemáticos el cálculo de la mejor ruta a cada destino.
- Monitorear constantemente la red para detectar cualquier cambio en la topología que puedan representar rutas conocidas que ya no sean válidas.

4.2. Tipos de Ruteo.

4.2.1. Ruteo estático.

El ruteo estático es la forma más simple de encaminamiento ya que se trata de rutas programadas manualmente que en consecuencia serán fijas. Por lo que a un administrador le compete el cálculo de rutas, así como el encontrar las mismas, y propagarlas a través de la red. Los ruteadores una vez configurados no tienen necesidad de intentar descubrir otra ruta, en tiempo real de ejecución los cambios que se quieran aplicar en la red son nulos.

Las ventajas que ofrece este tipo de ruteo, es su simplicidad para aplicarlos en redes pequeñas, se le considera como un recurso eficaz ya que se utiliza mucho menos ancho de banda para servicios de transmisión, da una baja sobrecarga en otros ruteadores y no necesita gastar recursos del CPU del ruteador intentando calcular rutas lo que implica menos uso de memoria.
Al mencionar varias de éstas ventajas también se debe de tener en cuenta que este tipo de ruteo tiene varias limitaciones. Como puede ser que en caso de un fallo en la red, o un cambio en la topología original, es responsabilidad del administrador de la red hacer manualmente los cambios y ajustes correspondientes.

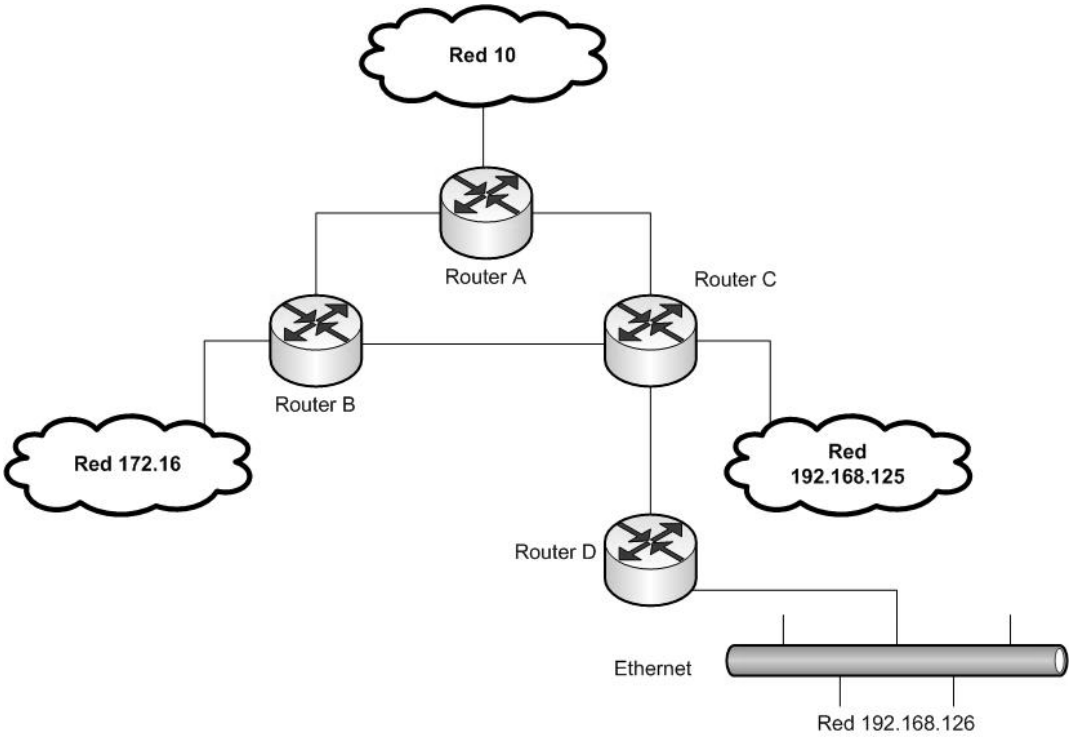


Figura 4.1 Representación de una pequeña topología de red.

En la figura 4.1 se puede ver como un administrador de red ha colocado en un esquema que considera el más eficaz, para cuestión de minimizar la carga y tráfico en la red. Esta red de trabajo es relativamente pequeña, ya que sólo consta de tres redes diferentes, una de ellas soporta una red interna. Cada red utiliza su propio espacio de direcciones y un protocolo de enrutamiento diferente; dada la incompatibilidad entre los protocolos de enrutamiento, se puede decidir que no redistribuye información de enrutamiento entre sus redes. En vez de esto, se pueden agregar rutas a los números de la red, y definir estáticamente caminos entre ellas. En la tabla 4.1 se puede ver la tabla de enrutamiento de los tres ruteadores. El ruteador D conecta una red interna pequeña a las otras redes, ese ruteador utiliza su puerto serie como Gateway predeterminado para todos los paquetes dirigidos a cualquier dirección IP que no pertenezca a 192.168.126.

Ruteador	Destino	Siguiente salto
A	172.16.0.0	B
A	192.168.0.0	C
B	10.0.0.0	A
B	192.168.0.0	C
C	10.0.0.0	A
C	172.16.0.0	B
C	192.168.126.0	D

Tabla 4.1 Tabla con ejemplo de rutas estáticas

En el ejemplo, el ruteador A envía todos los paquetes dirigidos a cualquier host del espacio de direcciones de la red 172.16 al ruteador B. El ruteador A también envía todos los paquetes dirigidos a los hosts de la red 192.168 al ruteador C. El ruteador B envía todos los paquetes dirigidos a cualquier host de la red 10 al ruteador A. El ruteador C envía todos los paquetes destinados a la red 10 al ruteador A, los destinados a 172.16 son enviados al ruteador B. Además, el ruteador C envía los paquetes dirigidos a 192,168.126 al ruteador D y a su red interna. Esta red es interna porque literalmente es un callejón sin salida de la red, ya que sólo hay un camino de entrada y uno de salida. Y lo que se puede ver es que esta red depende completamente del ruteador C y del propio ruteador C para tener conectividad con todos los hosts conectados a la inter-network.

Por lo que se hace evidente en este ejemplo que una de las desventajas, es cuando un enlace falla, ya que daría lugar a que haya destinos inalcanzables a pesar del hecho de que hay una ruta alternativa disponible para ser utilizada. Como se ve en la figura 4.2 cuando un enlace ha fallado.

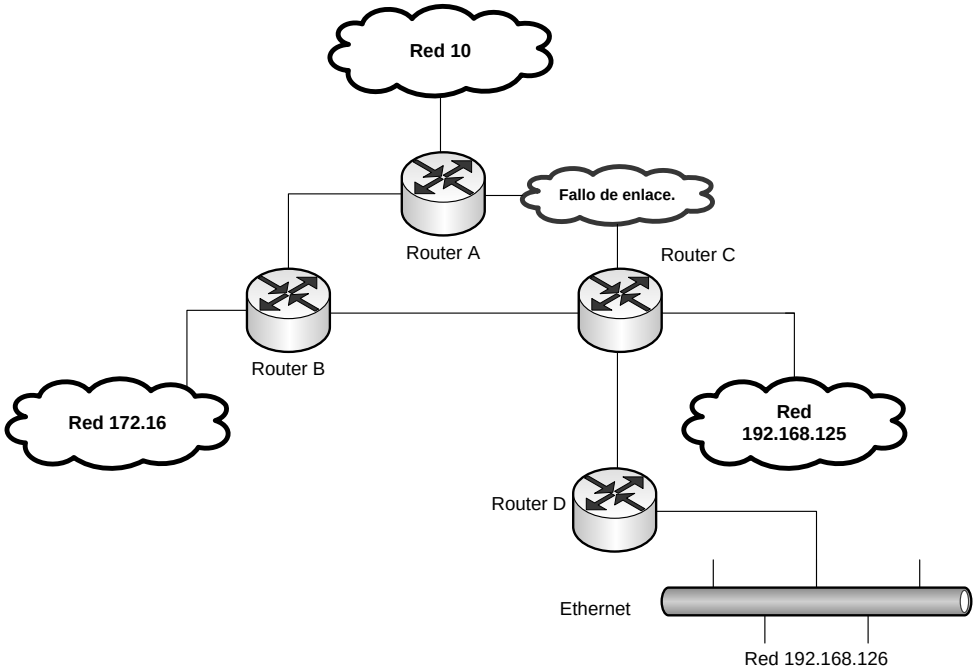


Figura 4.2 Ejemplo de fallo de uno de los enlaces que interrumpe las comunicaciones.

El efecto del fallo, es que los sistemas finales de las redes 10 y 192 no puedan comunicarse entre si, aun cuando existe una ruta válida a través del ruteador B. La tabla 4.2 muestra los efectos que tiene el fallo del enlace.

Ruteador	Destino	Siguiente salto
A	172.16.0.0	B
A	192.168.0.0	C- Inaccesible
B	10.0.0.0	A
B	192.168.0.0	C
C	10.0.0.0	A- Inaccesible
C	172.16.0.0	B

Tabla 4.2 Ruteadores inaccesibles por fallo en un enlace.

Por lo que en este tipo de ruteo los caminos son definidos y no tienen un mecanismo dinámico para que se puedan alterar, se impide que los ruteadores A y C reconozcan el fallo del enlace, dado que no utilizan un protocolo de enrutamiento que realice el cambio; de otro modo descubrirían y probarían las cualidades de los otros enlaces para conocer los destinos. En consecuencia no podrán descubrir la ruta alterna por el ruteador B, que es válida, hasta que el administrador realice manualmente una acción para corregirlo.

4.2.2. Ruteo Dinámico.

- **Ruteo por vector de distancia**.
  Este mecanismo se basa en algoritmos de vector de distancia, algunas veces llamados algoritmos de Bellman-Ford, los cuales transmiten periódicamente copias de sus tablas de enrutamiento a sus vecinos de la red inmediata. Cada receptor añade un vector de distancia (es decir, su propio "valor" de distancia) a la tabla y lo envía a sus vecinos inmediatos. Esto se produce de un modo omnidireccional entre los ruteadores que son vecinos inmediatos. Este proceso paso a paso hace que cada ruteador aprenda sobre otros ruteadores y desarrolle una perspectiva acumulativa de las distancias de la red.

  La tabla acumulativa se utiliza entonces para actualizar las tablas de enrutamiento de los ruteadores. Una vez completa, cada ruteador ha aprendido una vaga información sobre las distancias a los recursos conectados a las redes.

  Una de las desventajas del enrutamiento por vector de distancia es que un fallo o cualquier otro cambio en la red, implicará algún tiempo de convergencia para que los ruteadores puedan tener un nuevo entendimiento de la topología de red. Durante el proceso de convergencia, la red puede ser vulnerable al enrutamiento incoherente y enlaces inaccesibles, e incluso caer en bucles infinitos, por lo que el rendimiento de la red está en riesgo durante el proceso. De tal forma que los protocolos más antiguos de vector de distancia no son apropiados para redes WAN grandes y complejas.

La primera versión de este protocolo de enrutamiento fue uno de los primeros Protocolos de Gateway Interior, IGP por sus siglas en inglés (Interior Gateway Protocol) que se utilizaron en redes IPv4; RIP por sus siglas en inglés (Routing Information Protocol), es un protocolo de vector de distancia basado en el algoritmo Bellman-Ford y busca una ruta óptima mediante el conteo de saltos, considerando cada ruteador que atravesar para llegar a su destino, además de ser un protocolo abierto, es decir, no se tiene que tener una licencia de un fabricante para su uso.

El protocolo de ruteo de información, RIP, utiliza como métrica de enrutamiento el conteo de saltos, no toma en cuenta otros parámetros importantes en la elección de la ruta como por ejemplo, el ancho de banda o el tráfico del enlace. RIP utiliza UDP para publicar información de enrutamiento hacia otros ruteadores con RIP.
Este protocolo fue diseñado para trabajar con IPv4 e IPX en pequeñas redes, pero tiene limitaciones importantes:

- *La escalabilidad* de este protocolo de enrutamiento está limitado a un rango de 15 saltos máximo.
- *La métrica de conteo de saltos* resulta inadecuada para elegir rutas que dependan de parámetros en tiempo real, como por ejemplo, los retardos o la carga de enlace.
- *La velocidad de convergencia* de RIP se considera lenta comparada con otros protocolos de enrutamiento de estado enlace.

Cada host de la red que utiliza el protocolo de enrutamiento RIP tiene los siguientes campos.

- *Dirección destino*. Contiene la dirección de la red final a la que se desea acceder y tendrá que ser obligatoriamente “classfull”, el término indica que la red debe tener en cuenta la clase y por lo tanto no deberá ser compuesta por subredes.
- *Siguiente salto*. Se define como el siguiente ruteador que tiene que atravesar un paquete para llegar a su destino, el siguiente salto será necesariamente un ruteador vecino del ruteador origen.
- *Interfaz de salida del ruteador.* La interfaz de salida que está directamente conectada al siguiente salto.
- *Métrica.* El conteo de saltos, se considera cada uno como único, independiente de otros factores como tipo de interfaz o tráfico del enlace. La métrica total consiste en el total de saltos desde el ruteador origen hasta el ruteador destino, con la limitación de que 16 saltos se consideran como destino inaccesible, esto limita el tamaño de la red.
- *Temporizador.* El temporizador indica el tiempo transcurrido desde que se ha recibido la última actualización de una ruta.

RIPv2 incorpora el siguiente conjunto de mejoras respecto a RIPv1:

- Autenticación para la transmisión de información RIP entre vecinos.
- Utilización de máscaras de subred, por lo que el mecanismo de máscaras de subred de tamaño variable, VLSM por sus siglas en inglés (variable

length subnet mask) es viable.
- Utilización de máscaras de subred en la elección del siguiente salto, con lo que se permite arquitecturas de redes discontinuas.
- Utilización de la dirección multicast IPv4 para el envío de actualizaciones de tablas RIP.

A pesar de las mejoras, RIPv2 tiene algunas limitantes importantes:
- Limitación en el tamaño máximo de la red, debido a que el número de saltos máximo como en RIPv1 sigue siendo 15 saltos, por lo que en redes grandes el uso de RIPv2 no es recomendado.
- RIPv2 es un gran generador de tráfico.
- RIPv2 sólo permite una ruta por cada destino, así que no se puede realizar balanceo de carga.

- **Enrutamiento por estado de enlace.**

Los algoritmos de enrutamiento por estado de enlace o conocidos como la primer ruta más corta, SPF por sus siglas en inglés (Shortest Path First), mantienen una base de datos compleja de la topología de red. A diferencia del protocolo de vector de distancia, los protocolos de estado de enlace desarrollan y mantienen un conocimiento completo de la red, y del modo en que se interconecta. Esto lo realizan mediante el intercambio de publicaciones del estado del enlace, LSA por sus siglas en inglés (Link-State Advertisement) con otros ruteadores de la red.
Cada ruteador que ha intercambiado LSA constituye una base de datos topológica usando todas las LSA recibidas. Es cuando se utiliza un algoritmo SPF para calcular la accesibilidad de los destinos en la red. Esta información se usa para actualizar la tabla de enrutamiento. Este proceso puede descubrir cambios en la topología de la red, debido a fallos de un componente o al crecimiento de la red. De hecho, el intercambio de LSA es disparado por un evento en la red, en lugar de ejecutarse periódicamente, de tal forma que no hay necesidad de esperar a que expiren una serie de temporizadores arbitrarios para que los ruteadores de la red puedan empezar a converger.

A pesar de todas sus características y flexibilidad, el enrutamiento por estado de enlace tiene dos riesgos potenciales:
- Durante el proceso de descubrimiento inicial, los protocolos de enrutamiento por estado de enlace pueden inundar los servicios de transmisión de la red y por lo tanto reducir significativamente la capacidad de la red para transportar datos. El efecto de bajo rendimiento es temporal, pero puede ser muy notable.
- Este enrutamiento es muy intensivo en cuanto a memoria y procesamiento, por lo que se necesitan ruteadores mejor adaptados para soportar una carga de trabajo mayor, que si se tratará del enrutamiento por vector de distancia. Lo que hace que el costo de los ruteadores para el enrutamiento por estado de enlace sea más elevado.

En una red bien planificada, un protocolo de enrutamiento por estado de enlace le permite a la misma solucionar fácilmente los efectos de un cambio de topología repentino, por lo que el estado de enlace puede ser muy útil en redes de cualquier tamaño.

- **OSPF**

  La primer ruta abierta más corta, OSPF por sus siglas en inglés (Open Shortest Path First) es un protocolo que se utiliza para las redes de tamaño medio a grande, debido a que permite una escalabilidad muy accesible y eficaz. Entre sus muchas características es destacable que no tiene el problema de limitación de los 15 saltos de RIP, además de que los tiempos de convergencia de OSPF son mejores en todos los casos. OSPF toma en cuenta factores como el ancho de banda para el cálculo de costos y redes óptimas.

  Este protocolo utiliza el algoritmo de estado enlace, los ruteadores de estado enlace mantienen una imagen común de la red e intercambian su información de enlaces desde un descubrimiento inicial hasta los cambios de la red. Estos equipos no realizan broadcast de sus rutas periódicamente como los ruteadores que utilizan vector distancia. OSPF tiene las siguientes características:

  - *Velocidad de convergencia*. OSPF tiene un tiempo de convergencia mucho menor que el protocolo RIP, ya que sólo se actualizan las rutas que han sido modificadas y se distribuyen a través de la red de forma rápida.
  - *Soporta el mecanismo VLSM*
  - *Tamaño de la red*. OSPF no tiene la limitante de 15 saltos para alcanzar determinado destino. Es un protocolo adecuado para implementarse en redes de mayor tamaño. Por poner un ejemplo, para OSPF se recomienda que las áreas donde se implementa no deban contener un número superior a los 400 ruteadores.

  A diferencia de RIP que inunda la red con broadcast, OSPF contiene la tabla de enrutamiento de la red actualizada, con un período de aproximadamente cada 30 segundos, por lo que el protocolo envía actualizaciones cuando se produce un cambio en la red, con ello hace un mejor uso del ancho de banda. OSPF selecciona una ruta, la cual utiliza una métrica basada en el ancho de banda y los retardos del enlace.

  La agrupación de miembros, que es utilizada por RIP como una topología plana en la cual todos los ruteadores forman parte de la misma red, provoca que la comunicación entre ellos tenga que navegar por la totalidad de la red, de ésta forma cada cambio en un ruteador afectaría al resto de los equipos de la red. Por su parte, OSPF introduce el concepto de “áreas” lo que permite la segmentación de la red en porciones más pequeñas. Un área es un conjunto de redes dentro de un sólo Sistema Autónomo, AS por sus siglas en inglés (Autonomous System) que se han agrupado. La topología de un área permanece ocultar al resto del sistema autónomo, y cada área tiene una base

de datos topológica separada. El enrutamiento en el AS se produce en dos niveles, dependiendo de si la fuente y el destino de un paquete están en la misma área (enrutamiento intra-área) o en áreas diferentes (enrutamiento Inter- áreas).

El enrutamiento intra-área lo determina sólo la propia tecnología del área. El paquete se encamina a partir de información obtenida dentro del área, no se puede utilizar información obtenida fuera de la misma.
El enrutamiento en una red interna se hace a través del backbone.

4.2.3. Ruteo IPv6 vs Ruteo IPv4.

Uno de los aspectos importantes que se tiene en consideración para el ruteo IPv6 es la distancia administrativa, que es un valor que representa la confiabilidad del protocolo de enrutamiento. Durante el proceso de envío, los ruteadores utilizan la distancia administrativa para seleccionar la mejor ruta, cuando múltiples rutas utilizan diferentes protocolos de enrutamiento que apuntan hacia el mismo destino de red. La distancia con menor valor, es la ruta prioritaria para el ruteador. Las distancias administrativas de los protocolos de enrutamiento soportadas por IPv6 no fueron modificadas en los protocolos equivalentes en IPv4. Un ejemplo de este tipo de distancias es la Tabla 4.3 donde se muestran los valores de algunos protocolos de enrutamiento.

Protocolo de enrutamiento	Distancia administrativa
Interfaz conectada	0
Ruta estática (hacia la interfaz)	0
Ruta estática (hacia el siguiente salto)	1
BGP externo (eBGP)	20
OSPF	110
IS-IS	115
RIP	120
BGP Interno (iBGP)	200

Tabla 4.3 Distancias administrativas de los protocolos usados en IPv4 e IPv6

Los protocolos de ruteo de información como RIP, Sistema Intermediario-Sistema Intermediario IS-IS por sus siglas en inglés (Intermediate System to Intermediate System), la primer ruta abierta más corta, OSPF y la mejora del protocolo de enrutamiento de Gateway Interior, EIGRP por sus siglas en inglés (Enhanced Interior Gateway Routing Protocol), soportan IPv6 por lo que a continuación se describen las versiones de éstos protocolos.

RIPng.

El Protocolo de Ruteo de Información con soporte IPv6, RIPng por sus siglas en inglés (Routing Information Protocol next generation) basado en algoritmos de vector distancia conocido como Bellman-Ford, para este protocolo muchos conceptos fueron tomados de RIPv1 y RIPv2; al igual que éstos, RIPng continua limitado a un radio de operación de 15 saltos o del tiempo de convergencia, que puede ser demasiado grande. RIPng utiliza datagramas UDP para enviar y recibir información de encaminamiento. El broadcast periódico que contiene la información de encaminamiento es enviado utilizando direcciones multicast para reducir tráfico en nodos que no requieren escuchar los mensajes RIP.

Los cambios hechos para RIP se muestran a continuación:

- La longitud de los prefijos de las direcciones destino, así como las direcciones del siguiente salto se adjuntan a la longitud de una dirección IPv6 a 128 bits.
- Los mensajes RIPng se envían sobre paquetes IPv6.
- El número de puerto estándar de UDP para IPv6 es el 521 en lugar del 520 que era destinado para direcciones IPv4, por lo que este puerto envía y recibe información de enrutamiento entre ruteadores RIPng.
- Las actualizaciones se envían entre los ruteadores RIPng vecinos, usando la dirección de enlace local FE80::/10 como dirección origen.
- La dirección multicast estándar usada con RIPng es FF02::5, en lugar de la 224.0.0.9 en IPv4. La dirección FF02::9 representa la dirección multicast para “todo los ruteadores” en un ámbito de enlace local.

IS-IS

El protocolo de Sistema Intercambio-Sistema Intermediario, IS-IS por sus siglas en inglés, fue originalmente diseñado como un protocolo de enrutamiento OSI, que posteriormente fue adaptado para poder soportar a IPv4. IS-IS es un protocolo de estado enlace basado en el algoritmo Dijkstra, posee una gran escalabilidad y además es jerárquico. Los ruteadores IS-IS deben ser miembros de áreas IS-IS. También proporciona la información del costo de los enlaces (el costo por defecto de una interfaz es 10) para calcular la mejor ruta y alcanzar las redes destino.
El tiempo de convergencia que ofrece IS-IS es notablemente menor en comparación al tiempo de convergencia de RIP. Las características de este protocolo lo hacen ideal para soportar IPv6 aunque para esto se requirió hacer una serie de modificaciones debido a que la nueva versión del protocolo representa una nueva familia de direcciones que soportar.

Las mejoras a este protocolo se enfocan en dos nuevos Tipos de Valor de Longitud, TLV por sus siglas en inglés (Type Lenght Values), los cuales fueron añadidos para transportar información relacionada con el encaminamiento IPv6. El TLV es información de ruteo codificada en campos de longitud variable dentro de los paquetes de estado enlace. Los nuevos TLVs añadidos son:

- *Accesibilidad IPv6*. Este nuevo TLV describe la accesibilidad de la red en términos del prefijo de enrutamiento IPv6, información de métrica y algunos bits de opción. Los bits de opción indican la publicación del prefijo IPv6 desde un nivel más alto, distribución del prefijo desde otros protocolos de enrutamiento (redistribución) y la existencia de sub TLVs. El valor decimal asignado al TLV de accesibilidad IPv6 es 236 (hex 0xEC).
- *Dirección de interfaz IPv6*. Este TLV contiene una dirección de interfaz de 128 bits en lugar de una dirección de interfaz IPv4 (32 bits). El valor decimal asignado al TLV es 232 (hex 0xE8).

El protocolo IS-IS está basado en una estructura de dos niveles. Cualquier ruteador IS-IS puede desempeñar los siguientes roles:
Ruteador de nivel 1 (L1) Es el responsable del encaminamiento IPv6 y son llamados sistemas intermediarios.
Ruteador de nivel 2 (L2). Es el responsable del encaminamiento IPv6 y son llamados ruteadores centrales.

También se deben de tener en cuenta algunas consideraciones debido a que al diseñar redes con IS-IS se tendrán las dos versiones, IPv4 e IPv6 trabajando juntas.

Consideración para ruteadores IS-IS IPv6 adyacentes: Se pueden tener tres posibles arquitecturas de ruteadores IS-IS adyacentes en un área IS-IS:

- IS-IS sólo con IPv4.
- IS-IS sólo con IPv6.
- IS-IS con IPv4/IPv6.

Configuración de ruteador nivel 2 para IPv6: Ruteador de nivel 2 IS-IS es el responsable del intraruteo. Los ruteadores de nivel 2 deben ser contiguos con otros de la misma versión, de lo contrario se tendrán hoyos negros como resultado. Los hoyos negros son resultado de tener un ruteados IS-IS IPv4 en la ruta más corta de un ruteador IS-IS IPv6. Los ruteadores de frontera envían paquetes a un ruteador IS-IS IPv4 pero éste no sirve como un siguiente salto debido a que no entenderá las direcciones IPv6.

Multitopología para IPv6: Una alternativa a las restricciones del diseño de las redes IS-IS es cuando IPv4 e IPv6 son habilitados e implementados, y cuentan con un algoritmo SPF para cada familia de direcciones. En este caso cada versión puede manejar una topología por separado.

OSPFv3

El Protocolo de la primer ruta abierta más corta, OSPFv3 por sus siglas en inglés (Open Shortest Path First), esta definido en el RFC 5340 [21]. Es un protocolo complejo y extenso, con soporte IPv6, aunque muchas de sus especificaciones están basadas en OSPFv2, se le han realizado algunas mejoras. Después de haber actualizado el soporte para IPv6, OSPFv3 puede distribuir prefijos IPv6 y correr nativamente sobre IPv6.

OSPFv3 tiene algunas similitudes con OSPFv2 que son:

- OSPFv3 usa los mismos tipos de paquetes básicos que OSPFv2 como son hello, DBD (también llamado DDP, paquetes de descripción de la base de datos), LSU (actualización estado-enlace), LSR (petición estado-enlace) y LSA (anuncio estado-enlace).
- El mecanismo para el descubrimiento de vecino y adyacentes es idéntico.
- El envío y duración de los paquetes LSA son los mismos en OSPFv2 y OSPFv3.

Las actualizaciones importantes de OSPFv3 respecto a OSPFv2 se mencionan a continuación:

- *El protocolo* procesa por enlace no por subred. IPv6 conecta interfases a enlaces. Múltiples subredes IP pueden ser asignadas a un enlace simple y dos nodos pueden hablar directamente sobre un mismo enlace, incluso si no comparten una subred en común. OSPF para IPv6 funciona por un enlace en lugar de hacerlo por subred. Los términos "Red" y "Subred" usados en OSPFv2 pueden ser reemplazados con el término "Enlace"; por ejemplo, una interfaz OSPFv3 se conecta a un enlace en vez de una subred.
- *Las direcciones IPv6* ya no son presentadas en los encabezados de los paquetes de OSPFv3, éstas son presentadas como información de carga útil. Soporte explícito para múltiples casos por enlace. Las diferentes versiones de OSPF pueden funcionar sobre un mismo enlace. Una utilidad de está característica es el que un sólo enlace pertenezca a varias áreas.
- *Uso de direcciones de enlace local*. OSPFv3 asume que a cada interfaz le ha sido asignada una dirección unicast de enlace local (fe80::) así que utiliza estas direcciones para identificar a los vecinos adyacentes.
- *Las direcciones multicast* se especifican de la siguiente manera:
  FF02::5. Representa a todos los ruteadores que utilizan OSPF dentro de un enlace local. Esta dirección multicast es equivalente a 224.0.0.5 en OSPFv2.
  FF02::6. Representa a todos los ruteadores designados (DR) en el enlace local. La dirección equivalente en OSPFv2 es 224.0.0.6.
- *Seguridad.* OSPFv3 utiliza los encabezados de extensión IPSec y ESP como mecanismos de autenticación en lugar de la variedad de esquemas de autenticación y procedimientos definidos en OSPFv2.
- *Transporte.* Los mensajes OSPFv3 son enviados sobre datagramas IPv6, permitiendo la configuración a través de túneles 6over4.

BGP

El protocolo ruteo exterior, BGP por sus siglas en inglés (Border Gateway Protocol) es usado para la conexión entre sistemas autónomos, AS por sus siglas en inglés (Autonomous Systems) y así poder hacer un intercambio de información de rutas entre éstos sistemas.
BGP es un protocolo de enrutamiento vector distancia que utiliza TCP en el puerto 179 para poder realizar conexiones con otros ruteadores BGP que son llamados vecinos BGP. La información relacionada con el vector distancia por BGP entre vecinos es llamada *atributo.*

BGP intercambia información de las redes a las cuales puede acceder con los vecinos usando mensajes de actualización. Esos mensajes incrementan cuando las actualizaciones son intercambiadas entre vecinos BGP. Si un ruteador es agregado o removido, un mensaje de actualización es enviado para informar a los vecinos.
Durante esta operación en una red de redes, un ruteador BGP tiene múltiples sistemas autónomos y diferentes rutas para acceder a una red destino en particular. El algoritmo BGP designa la mejor ruta a través de los sistemas autónomos para acceder a una red en específico de una lista de caminos posibles. La determinación de las rutas de los sistemas autónomos esta hecha en base a la lista de atributos. BGP fue diseñado para ser un protocolo altamente escalable y poder utilizarse en grandes redes como la Internet global.

En el RFC 4271 [22], se definen las normas para este protocolo, así como las implementaciones y usos que se le da hoy en día, pero esto sólo es aplicado al encaminamiento para IPv4.

Se creó una versión mejorada llamada BGP4+, mejor conocida como multiprotocolo BGP, que extiende sus especificaciones, para incluir múltiples extensiones al protocolo como nuevas familias de direcciones, IPv6, IPX y VPN. De está forma BGP4+ puede llevar información de encaminamiento para IPv6 y otros protocolos incluido IPv4. El RFC 4760 [23], define los atributos que fueron actualizados para el manejo de direcciones IPv6 con BGP.

Algunos atributos que se actualizaron en las especificaciones de BGP para que pudiera soportar IPv6 son:

- **Siguiente salto (Next_Hop):** Este atributo define la dirección del ruteador de frontera que deberá ser usado como el siguiente salto de destino. El atributo de Siguiente salto es expresado como una dirección IPv6. Éste puede contener una dirección IPv6 unicast global o el par de direcciones unicast global y Enlace local IPv6 del *siguiente salto*.
    - **Las direcciones de Unicast IPv6:** Como se mencionó en el capítulo 2, las direcciones tienen el prefijo 2000::/3, por ejemplo, un *siguiente salto* puede tomar un valor de 2001:448:11:1::1 y puede ser utilizada como una dirección unicast global.
    - **La dirección IPv6 enlace local:** Como se mencionó en el capítulo 2, las direcciones tienen el prefijo fe80::/10. Una dirección de enlace local puede ser usada como dirección de *siguiente salto* con BGP4+ y éstas podrán ser accesibles por los vecinos BGP. Por ejemplo, para un *siguiente salto* a buscar un ruteador vecino BGP4+ se usa una dirección de enlace local que podría ser fe80::200:abcd:af56:fefc.
    - **Información de alcance del nivel de red**, NLRI por sus siglas en inglés (Network Layer Reachability Information) es un valor de destino. Un destino es definido en BGP como un prefijo de red con el valor de la longitud del prefijo. Este atributo puede ser expresado como un prefijo IPv6 con BGP4+.

Capítulo 5
DNS en IPv6

5.1. Introducción.

Una de las formas más comunes para un usuario de referirse a una dirección de Internet, es a través del uso de literales o de un nombre, ya que no sería fácil recordar direcciones numéricas de 32 bits y sería aun más difícil recordar la forma hexadecimal de una dirección IPv6, cuya estructura y longitud se ha mencionado en capítulos anteriores. Hoy en día se utilizan comúnmente dos esquemas para transferir las direcciones de su forma numérica a los nombres que son más fáciles de recordar y éstos son: El Servicio de Nombre de Dominio, DNS por sus siglas en inglés (Dominian Name Service), y el Servicio Información sobre la Red, NIS por sus siglas en inglés (Network Information Service), que ahora es parte del sistema de archivos de red, NFS por sus siglas en inglés (Network File System).

Servicio de Nombre de Dominio.
Un nombre simbólico, es una cadena de caracteres que se utilizan para identificar una máquina y éste puede ser sencillo o tan complejo como se quiera, ya que en el nombre se puede tener una pequeña descripción del equipo al que se le está asignando ese nombre. Cuando se quiera enviar información hacia una máquina remota, se debe de utilizar su dirección IP. En vez que el usuario memorice los números del equipo remoto, es más común utilizar un nombre, después de todo es más fácil de recordar que una dirección de 32 bits o una de 128 bits.

En un principio, el querer convertir un nombre simbólico a una dirección IP podía llevarse a cabo en cada equipo, dado que se puede tener un registro de dichas direcciones con su nombre simbólico asociado, como se da el caso del archivo host que contiene cada equipo, pero esta aproximación solamente puede trabajar de forma correcta cuando se encuentra dentro de una red pequeña, ya que en redes muy grandes como es el caso de Internet donde diariamente hay incorporaciones y modificaciones a los nombres existentes, el tiempo requerido para actualizar cada equipo de las redes sería enorme.

Es por eso que una solución al problema ha sido el ofrecer un método que permita sacar de los Centros de Información de la Red, NIC por sus siglas en inglés (Network Information Center) que rigen Internet, la administración de las tablas de consulta y llevarla a los participantes y a sus redes autónomas, de tal forma que la carga de cada red sea pequeña, pero que a la vez no se comprometa la flexibilidad. Esto es lo que hace un DNS, que a través de procesos que corren en un equipo llamado servidor de nombres y otro equipo maneja la definición de nombres simbólicos utilizando métodos del DNS, para que así parte del sistema sea una biblioteca de funciones que se pueden utilizar en las aplicaciones para llevar a cabo consultas sobre el servidor de nombres. A estas rutinas de consulta se les llama *definidor* (resolver) o *definidor de nombres.*

El *servicio de nombres de dominio* como su nombre lo indica, funciona dividiendo la red global en un conjunto de dominios, o redes, que pueden a su vez dividirse posteriormente en subdominios. Esta estructura tiene una jerarquía que asemeja a un árbol como se muestra en la figura 5.1, donde aparecen algunos nombres de dominio existentes. El primer

conjunto de dominios se llama dominios de alto nivel y a continuación se listan algunos que se encuentran en uso:

- ARPA: para organizaciones que se refieren específicamente a Internet.
- COM: para empresas comerciales.
- EDU: para organizaciones educacionales.
- GOV: para organismos gubernamentales.
- MIL: para organizaciones militares.
- ORG: para organizaciones no comerciales.

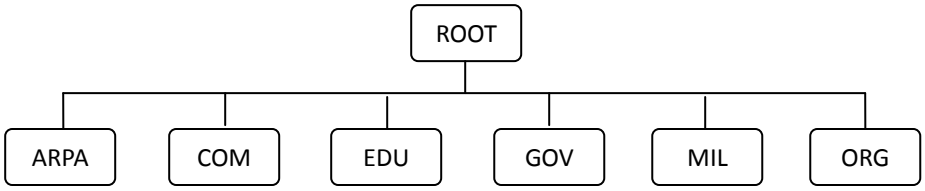


Figura 5.1 Estructura de dominios de Internet.

Además de los dominios de alto nivel existen dominios especializados de alto nivel para cada país que está conectado, éstos generalmente se identifican por una abreviatura del nombre del país como mx para México; estas abreviaturas se dejan fuera de los diagramas de la estructura de Internet por conveniencia. Por debajo de los dominios de alto nivel existen otros para las organizaciones individuales, que se encuentran dentro de cada dominio de alto nivel. Todos los nombres de dominio están registrados en un NIC y son exclusivos para la red.
Generalmente los nombres son representativos de las compañías y las organizaciones. Existen dos formas de nombrar un destino. Si el destino está en la red global se utiliza el nombre absoluto, el cual es único, entendible y especifica el dominio de la máquina destino. Si el nombre relativo puede utilizarse, ya sea dentro del dominio local, donde el servidor de nombres sabe que el destino está dentro del dominio, y por lo tanto, no necesita encaminar la información hacia la red global, o cuando el nombre relativo lo conoce el servidor de nombres, puede ampliarlo y encaminarlo correctamente.

Cada servidor de nombre administra un área diferente de red, el conjunto de máquinas administradas por el servidor de nombres se llama *zona.* Pueden administrarse varias zonas por un mismo servidor de nombres por lo que es muy común que cada zona tenga un servidor de nombres de respaldo o secundario, donde los dos servidores de nombres (primario y secundario) contienen la misma información. Los servidores de nombres que se encuentran dentro de una zona se comunican utilizando un *protocolo para transferencia de zonas.*

DNS opera teniendo un conjunto de zonas anidadas ya que cada servidor de nombres se comunica con la zona que se encuentra por encima de él, y si tiene alguna por debajo de también tendrá comunicación. Cada zona tiene por lo menos un servidor de nombres que es responsable de conocer la información de la dirección de cada equipo que se encuentra dentro de esa zona, así cada servidor de nombres también utiliza el Protocolo Datagrama de Usuario, UDP por sus siglas en inglés (User Datagram Protocol), puesto que su método de conexión brinda mejor desempeño, sin embargo, TCP se utiliza para las actualizaciones de la base de datos debido a su confiabilidad.

Cuando una aplicación de un usuario necesita definir un nombre simbólico dentro de una dirección de red, la aplicación envía una consulta al proceso definidor, el cual comunica la consulta al servidor de nombres. El servidor de nombres verifica sus propias tablas y regresa la dirección de la red que corresponde al nombre simbólico. Si el servidor de nombres no tiene la información que necesita puede enviar una solicitud a otro servidor de nombres. Este proceso se muestra en la figura 5.2, tanto los servidores de nombres como los definidores utilizan las tablas y los caches de la base de datos para mantener la información acerca de las máquinas que se encuentran en la zona local, así como la información que se solicitó recientemente desde el exterior de la zona.

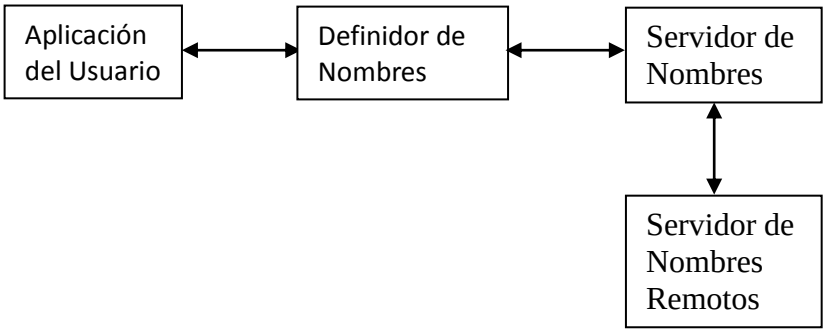


Figura 5.2 Forma del definir los nombres simbólicos.

Cuando un servidor de nombres recibe una consulta, las operaciones que puede realizar el definidor de nombres pueden clasificarse en:

- Recursivas
- No recursivas.

Una operación *recursiva* es aquella en la cual el servidor de nombres debe tener acceso a otro servidor de nombres para obtener la información.
Las *no recursivas* realizadas por el servidor de nombres incluyen una respuesta completa a la solicitud del definidor, una referencia hacia otro servidor de nombres o un mensaje de error.

Los registros de recursos, son la información que se necesita para definir un nombre simbólico que mantiene el servidor de nombres dentro de un conjunto de *registros de recursos*, que son entradas de una base de datos. Los registros de recursos (abreviados como RR) contienen información en formato ASCII. El formato de los registros de los recursos se muestra a continuación en la tabla 5.1.

Nombre (longitud variable)
Tipo (16 bits)
Clase (16 bits)
TTL (32 bits)
Longitud de los datos (16 bits)
Datos (longitud variable)

Tabla 5.1 Formato del registro de recursos.

Los campos de los registros de recursos se describen a continuación:
Nombre: El campo de Nombre es el nombre de dominio de la máquina a la que se refiere el registro, si no hay un nombre especificado se sustituye el nombre previamente utilizado.
Tipo: Es el campo que identifica el tipo de registro del recurso, los registros de recursos se utilizan para varios fines como mapear los nombres para las direcciones y definir las zonas. Los códigos utilizados en este campo se muestran en la tabla 5.2.

Número	Código	Descripción
1	A	Dirección de la red
2	NS	Servidor autorizado de nombres
3	MD	Destino del correo: ahora reemplazado por MX
4	MF	Enviador del correo: ahora reemplazado por MX
5	CNAME	Nombre alias canónico
6	SOA	Comienzo de una autoridad de zona
7	MB	Nombre del dominio para buzón
8	MG	Miembro del buzón
9	MR	Dominio para renombre de correspondencia
10	NULL	Registro nulo de recurso
11	WKS	Servicio bien conocido
12	PTR	Puntero hacia un nombre de dominio
13	HINFO	Información sobre host
14	MINFO	Información sobre el buzón
15	MX	Intercambio de correspondencia
16	TXT	Cadena de texto
17	RP	Persona responsable
18	AFSDB	Servicios del tipo AFS
19	X.25	Dirección X.25
20	ISDN	Dirección ISDN
21	RT	A través de ruta

Tabla 5.2 Tipos de registros de recursos.

Clase: El campo clase está dentro de la organización del registro de recursos y contiene un valor para la clase de registro. Los servidores de nombres de Internet generalmente tiene el código IN.

Tiempo de vida (TTL): Especifica el tiempo en segundos que valida el registro de recursos dentro del cache. Si se utiliza un valor de 0, el registro no debe agregarse al cache. Generalmente este campo indica al servidor de nombres cuánto tiempo tiene validez el registro antes de que tenga que pedir una actualización.

Longitud: Este campo especifica la longitud de la sección de los datos, es de longitud variable que describe de alguna forma la entrada. El uso de este campo varía con los diferentes tipos de registro de recursos. Algunos tipos de registros de recursos tienen un solo dato en el área de datos, como puede ser una dirección o un máximo de tres datos. El contenido de las áreas de datos del registro de recursos se presenta en la tabla 5.3

Tipo de RR	Campos del área de datos
A	Address: Una dirección
NS	NSDNAME: El nombre de dominio del host
MG	MGNAME: El nombre de dominio de un buzón
CNAME	CNAME: Un alias para la máquina
HINFO	CPU: Una cadena que identifica el tipo de CPU OS: Una cadena que identifica el sistema operativo
MINFO	RMAILBX: Un buzón responsable de enviar las listas EMAILBOX: Un buzón para mensajes de error
MB	MADNAME: Ahora obsoleto
MR	NEWNAME: Renombrar la dirección de un buzón especifico
MX	EXCHANGE: El nombre de dominio del host que actúa como intercambio de correspondencia
NULL	Se puede colocar cualquier cosa en el campo de datos
PTR	PTRDNAME: Un nombre de dominio que actúa como un puntero para una ubicación
TXT	TXTDATA: Cualquier tipo de texto descriptivo
WKS	Address: Una dirección de red Protocol: El protocolo utilizado Bitmap: Utilizado para identificar los puertos y los protocolos

Tabla 5.3 Contenido de las áreas de datos del registro de recursos.

El formato del registro de Inicio de Autoridad, SOA por sus siglas en inglés (Start of Authority), se utiliza para identificar las máquinas que se encuentran dentro de una zona. Hay un solo registro SOA en cada zona, el formato del campo de datos SOA se muestra en la tabla 5.4.

Nombre de dominio (MNAME)
Nombre del responsable (RNAME)
En serie
Tiempo de Refresco
Tiempo de Reintento
Tiempo de Vencimiento
Tiempo Mínimo

Tabla 5.4 Formato del registro de recursos SOA

Los campos de los registros SOA se describen a continuación:

El campo ***MNAME,*** es el nombre de dominio de la fuente de datos de la zona.
El campo ***RNAME,*** es el nombre de dominio correspondiente al buzón del administrador de la zona.
El campo ***En Serie,*** contiene un número de versión de la zona, este se incrementa cuando se cambia la zona, de otro modo, se mantienen con el mismo valor todos los mensajes.
El *tiempo de Refresco,* es el número de segundos que transcurre entre las restauraciones de los datos para la zona.
El *tiempo de Reintento,* es el número de segundos que hay que esperar entre las solicitudes de restauración que no tuvieron éxito.
El *tiempo de Vencimiento,* es el número de segundos que transcurren para que la zona de información ya no sea válida.
El *tiempo Mínimo,* es el número de segundos que van a utilizarse en el campo Tiempo de Vida de los registros de recursos dentro de la zona.

Los registros de recursos para la dirección consisten en el nombre de la máquina, el indicador del tipo de recurso y las direcciones de red por ejemplo un registro de recurso para la una dirección se vería de la siguiente forma:

TPCI_SCO_3 IN A 132.240.25.7
IN etiqueta el registro como una clase de Internet
IN-ADDR-ARPA

Los campos de direcciones, como sucede con el tipo de registro de recursos Address, utilizan un formato especial llamado IN-ADDR-ARPA. Este permite el mapeo inverso desde la dirección hacia el nombre del host, así como el mapeo del host a la dirección. Para entender a IN-ADDR-ARPA es útil comenzar con un registro de recursos en formato estándar. Uno de los tipos de registros de recurso más sencillo es el de la dirección (tipo A):
TPCI_SCO_3 IN A 132.240.25.7
TPCI_SCO_4 IN A 132.240.25.8
TPCI_SCO_5 IN A 132.240.25.9

Donde cada línea representa un registro de recursos. En este caso, todas son entradas que tienen el nombre simbólico del equipo, la clase del equipo (IN para Internet), A, para mostrar que se trata de un registro de recurso Address y la dirección de Internet.
La presentación de los registros facilita el mapeo del nombre hacia la dirección. El servidor de nombres simplemente busca una línea con el nombre simbólico que solicita la aplicación, y regresa la dirección de Internet que está al final de la línea.
La búsqueda desde la dirección hacia el nombre no es tan fácil, si los archivos de registro de recursos son pequeños, los retrasos que implica una búsqueda manual no se aprecian, pero en las zonas extensas pueden existir miles o decenas de miles de entradas. El índice de las bases de datos se respalda en el nombre, por lo que buscar una dirección puede ser un proceso lento. Para resolver este problema se elaboró el mapeo inverso IN-ADDR-ARPA, así para la información del registro de recursos del host, IN-ADDR-ARPA utiliza la dirección del host y un índice, cuando se localiza el registro de recursos adecuado puede extraerse el nombre simbólico.
IN-ADDR-ARPA utiliza el tipo de registro de recurso PTR para apuntar desde la dirección hacia el nombre.

En el RFC 1034 [27] fueron definidas las especificaciones en primera instancia para los DNS con IPv4, aunque estos RFC fueron actualizados en el RFC 1886 [28] para ampliar el servicio de direcciones con formato IPv6.
En el RFC 1886 [28] se define un nuevo tipo de registro denominado AAAA, que logra cumplir con el objetivo de llevar a cabo búsquedas en IPv6.
El problema que conllevan los sistemas DNS, se da por el hecho de que al hacer una consulta, las aplicaciones asumen que se les devolverá una dirección de 32 bits, para resolver este problema se deben definir:

- Un registro que mapee las direcciones de dominio con la dirección IPv6, lo lleva a cabo el registro AAAA mencionado.
- Un nuevo dominio que pueda soportar búsquedas basadas en direcciones IPv6. Este dominio es IP6.INT. Su representación se realiza en orden inverso de la dirección, separando los nibbles (hexadecimal) por puntos (“.”), seguidos de “.IP6.INT”. Así, la búsqueda inversa de la dirección 4321:0:1:2:3:4:567:89ab, seria “b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.0.1.2.3.4.IP6.INT”
- Redefinición de las consultas existentes, que localizan direcciones IPv4, para que puedan también procesar direcciones IPv6. Ello incluye todas las consultas, lógicamente (NS, MX, MB, ...)

Otro aspecto a considerar para el soporte de las direcciones IPv6, es la renumeración y el multi-homing, que incluye un nuevo tipo de registro el A6, que almacena las direcciones IPv6 de forma que se facilite la renumeración de la red.

5.2. Tipos de registros.

Registro AAAA.

El registro AAAA es un registro especificado en el RFC 1886 [28] el cual puede atender una sola dirección IPv6 de 128 bits, y el valor para este tipo de registro es 28 (decimal). Un host puede tener más de una dirección IPv6 por lo que también habrá registros AAAA para cada una de esas direcciones. Un registro tipo AAAA puede ser como:

www.example.com IN AAAA 2001:0:1:2:3:4:567:89ab

El dominio IP6.INT es definido para buscar un registro dada una dirección, la intención de este dominio es proporcionar una manera de relacionar una dirección IPv6 a un nombre de host, aunque puede ser usado para otros propósitos, el dominio está asociado a IP6.INT.

La búsqueda inversa para una dirección IPv6 se representa como el nombre de dominio IP6.INT por una secuencia de nibbles (un nibble está formado por 4 bits) separados por puntos con el sufijo ".IP6.INT". La secuencia de nibbles se codifican en orden inverso, es decir, el nibble de menor orden se codifica primero, seguido por el siguiente nibble de menor orden y así sucesivamente. Cada nibble se representa por un dígito hexadecimal.
El registro de búsqueda inversa es PTR de tipo 12. Por ejemplo, el nombre de dominio de búsqueda inversa correspondiente a la dirección

2001:0:1:2:3:4:567:89ab

Será:

b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.0.1.0.0.2.IP6.INT IN PTR www.ejemplo.com

Todos los tipos de consulta existentes que llevan a cabo procesamiento de sección adicional tipo A, es decir, los tipos de consulta como servidor de nombres (NS), intercambio de correo (MX) y buzón de correo (MB), se deben redefinir para llevar a cabo tanto el procesamiento mencionado como el procesamiento de sección adicional tipo AAAA. Estas nuevas definiciones significan que un servidor de nombres debe agregar cualquier dirección IPv4 e IPv6 relevante localmente disponibles a la sección adicional de una respuesta al procesar cualquiera de las consultas antes citadas.

A6

Es un tipo de registro ahora experimental, que se define en el RFC 2874 [29], del cual se sabe cuales pueden ser sus efectos en costo y riesgo por lo cual no se ha impulsado como el registro AAAA.
Las direcciones IPv6 se almacenan en uno o varios registros A6 y este tiene un valor de 38 (decimal). A un solo registro A6 se le puede incluir una dirección IPv6 o un conjunto de direcciones contiguas y la información conduce a uno o más prefijos. La información sobre el prefijo comprende su longitud y el nombre de dominio que es a su vez el propietario de uno o más registros A6 definiendo el prefijo o prefijos que se necesitan para formar uno o más direcciones IPv6 completas.
El formato de un registro A6 contiene dos o tres campos como se muestra en la figura 5.3.

Longitud del prefijo (1 octeto)	Sufijo de la dirección (0 – 16 octetos)	Nombre del prefijo (0-225 octetos)

Figura 5.3 Definición de campos del registro A6.

- La longitud del prefijo, está codificado como un entero de 8 bits con valor de entre 0 y 128, e indica la porción del sufijo de la dirección que es contenida en el registro.

- Un sufijo de dirección IPv6 codificado en orden de red (primer octeto de nivel superior). Debe tener octetos suficientes en este campo a un número de bits de la longitud del prefijo equivalente 128 o menos, de 0 a 7 bits de relleno para así hacer que este campo sea un número entero de octetos. Bits de relleno, si están presentes, deben ser puestos a cero cuando se carga un archivo de zona e ignorarlos.
- El nombre del prefijo, codificado como un nombre de dominio. Por las reglas de DNSIS, este nombre no puede ser comprimido.

El registro DNAME se describe en el RFC 2672 [30] y en conjunto con el registro A6 tienen soporte para la renumeración y agregación de direcciones IPv6. La combinación de los registros DNAME y A6 permite el mantenimiento del mapeo de direcciones a nombres cuando una red es renumerada o una unidad organizacional es renombrada. Por ejemplo, si una compañía cambia la NLA, todos los registros AAAA en el DNS necesitan ser cambiados. Con el nuevo registro A6, esto puede ser más sencillo. Un registro A6 puede incluir una dirección IPv6 completa o sólo una porción contigua de una dirección, como pueden ser los últimos 64 bits (que representan el ID de la interfaz) y ese hace referencia al resto de las direcciones por un nombre simbólico de dominio. La resolución o servidor de nombres seguirá la cadena de registro A6 del nombre de dominio del host al TLA ID.

La delegación de el espacio de direcciones con IPv6 no se logra a través de la zona de partición y el registro NS, pero basta con el registro DNAME. Similar al registro CNAME, que proporciona un alias al host, el registro DNAME proporciona un alias a una porción del nombre de un subárbol completo.

Por ejemplo cuando se busca resolver un nombre host.universe.com y se encuentra un registro DNAME de ejemplo.com DNAME venus.ejemplo.com. Este resuelve ejemplo.com a venus. ejemplo.com y busca host.venus.ejemplo.com.
Muchas aplicaciones en el mercado tienen soporte y usan el registro AAAA, dado que los registros DNAME y A6 no lograron pasar a la etapa de implementación amplia por sus requisitos complejos.

5.3. Representación de IPv6 en los servidores DNS.

Sin lugar a duda la ayuda de los DNS en la traducción de los nombres de las direcciones a un formato numérico y viceversa es de gran importancia, y no nos podríamos imaginar que este trabajo no se llevará acabo cuando se hace el uso de una red, debido a la gran información que se puede manejar o la infinidad de sitios que en nuestros días se acceden. Así que debido a esto se deben tener servidores de nombres con un soporte IPv6, para realizar las búsquedas de las direcciones IPv6 en el DNS, ya que las mismas serian imposibles de desplegar en su formato hexadecimal.

Desde 1995, en el RFC 1886 [28], se describe de forma clara como representar información IPv6 en los DNS, con esto se produjo un camino fácil para la mejora de los servidores sin embargo, en el 2000 se crea una mejora en el RFC 2874 [29], en el cual plantea un nuevo mecanismo para está implementación en particular, un análisis más detallado y la experiencia en la práctica mostró alrededor del 2001 que este método era muy ambicioso, por lo que el IETF comenzó a desecharlo y en 2003 el RFC 1886 [28] fue actualizado, lo que dio como resultado el RFC 3596 [24].

RFC 1886 AAAA y ip6.int

En este RFC se describe como las direcciones IPv4 son atendidas por el registro A, y el mapeo inverso está hecho para crear un nombre de dominio especial, el cual consiste en los valores de los octetos individuales de la dirección en un orden inverso, seguido por in-addr.arpa.

Así se realiza una búsqueda del nombre de dominio que como resultado en un registro PTR contiene el nombre de dominio asociado con la dirección en cuestión.
Por ejemplo así se puede representar una dirección IPv4 en un DNS:

```
www.ejemplo.com.        IN  A     192.0.2.17
17.2.0.192.in-addr.arpa  IN  PTR  www.ejemplo.com.
```

Los nombres terminan en un punto donde se indica que estas son las direcciones absolutas y que no se puede agregar un nombre de dominio adicional. El RFC 1886 [28], es muy puntual en la forma de cómo se harán las cosas para las direcciones IPv4, como es mencionado en el RFC 1034 [27]. Las direcciones IPv6 como ya se ha mencionado se almacenan en registros AAAA. El mapeo inverso está expresado por los dígitos hexadecimales de la dirección IPv6 (incluyendo todos los valores de la dirección, así como los ceros que normalmente se omiten en las diferentes representaciones de la dirección) en un orden invertido y agregando ip6.int.

Un ejemplo de cómo se representaría una dirección IPv6 de acuerdo con el RFC 1886 [28]es:

```
www.ejemplo.com      IN  AAAA      2001:db8:1bff:c001::390
0.9.3.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.c.f.f.b.1.8.b.d.0.1.0.0.2.ip6.int.  IN  PTR
www.ejemplo.com
```

Cuando el RFC fue publicado a mediados de 1990, el nivel de dominio .int era la infraestructura TLD de elección, ya que en ese tiempo era el único TLD reconocido internacionalmente; .com, .edu, .gov, .net, y .org todos eran consideraciones estadounidenses, y .arpa era relacionado a la desaparecida ARPANET.

RFC 2874: A6, DNAME Bitlabels, and ip6.arpa

Estos mecanismos no se encuentran en uso en el Internet actualmente ya que debido a que se encuentran en discusión varios problemas sobre IPv6, el IETF se enfoca en resolverlos y los relacionados con otros temas como ruteo y el tema de la renumeración rápida.

El registro A6, del nombre al mapeo de la dirección.

El RFC 2874 [29] habla acerca de la renumeración de la siguiente forma. Suponiendo que se pregunte por un número telefónico, como el de la oficina, se podría responder "1-555-555-5555." Con esto se podría comunicar con la oficina, mientras nada cambie. Aunque también es posible que responda "Extensión 224 al Laboratorio de redes emergentes", por lo que de cualquier forma se podrá llamar, con un esfuerzo extra en caso de que el número actual no este en funcionamiento, por ejemplo, que la extensión o el área de marcado del laboratorio cambie.
El RFC 1886 [28] AAAA describe un método similar a simplemente da a conocer el número solicitado. En cambio el RFC 2874 [29] dar un camino a seguir a través de la jerarquía de las direcciones, a diferencia de dar a conocer un número de teléfono proporciona una extensión, organización y ciudad.

Por ejemplo se puede ver la forma jerárquica del registro A6 como se encuentra en un DNS:

```
www          IN  A6   64  ::0000:0000:0000:0390 subnet-a
subnet-a     IN  A6   48  0000:0000:0000:c001:: prefix-isp1
subnet-a     IN  A6   48  0000:0000:0000:c001:: prefix-isp2
prefix-isp1  IN  A6   0   2001:0db8:1bff::
prefix-isp2  IN  A6   0   3ffe:9500:003c::
```

Los nombres de los dominios en el ejemplo son relativos, asumiendo que se encuentran en el archivo de zona de ejemplo.com, el servidor de nombre agrega .ejemplo.com después de cada nombre. Cada registro A6 provee una parte de la dirección y un apuntador a donde se encuentra el resto de la dirección para que pueda ser encontrada.

El primer registro A6 deja 64 bits en blanco para ser definidos más tarde y se continúa proporcionando una dirección IPv6 ::390 llenando los 128-64=64 bits como se especifique. Los bits especificados en la dirección en el registro A6 son copiados de la respectiva lista de direcciones. La parte de la dirección no está especificada así que debe de ponerse en ceros en la dirección que provee la zona de archivo. Así lo que se ve en el DNS es el nombre seguido por el valor del apuntador del registro A6 de la dirección IPv6, que le da un lugar en la jerarquía del DNS donde el resto de la dirección puede ser encontrada, en este caso bajo el nombre subnet –a. Y de hecho subnet –a es otro registro A6, o más bien, dos de ellos. Ambos establecen los bits entre 48 y 64 a c001 (el resto de los bits son cero) y el apuntador al prefix-1sp1 y prefix-isp2, respectivamente, para el resto de las direcciones.

Bajo esos nombres, los bits restantes del 0 a 48 son provistos. Y un apuntador a otro lugar no es necesario, porque ahora la dirección está completa. Como subnet-a tiene dos apuntadores para los 48 bits superiores, el procedimiento completo resulta en dos direcciones completas: 2001:db8:1bff:c001::390 y 3ffe:9500:3c:c001::390.
Con el registro A6, actualizando el DNS cuando este es un evento de renumeración pequeña, se tiene que hacer el cambio de todas las direcciones en todos los dominios para el sitio, que un simple registro A6 se puede hacer este cambio. Por ejemplo, si el prefijo 3ffe:9500:3c::/48 debe ser cambiado a 2007:4580:73::/48 esto sólo requiere una actualización del registro prefix-isp2, y todos los registros que apunten a este reflejaran la nueva información.

El Bitlabel y DNAME, de la dirección al mapeo de nombre.
Además del método A6 para transmitir el mapeo, el RFC 2874 [29] especifica una nueva forma de realizar el mapeo inverso de la dirección al nombre. Para esto se utilizan dos mecanismos que se encuentran definidos en los RFC´s 2672 [31] y 2673 [30], respectivamente: DNAME y bitlabels. El registro DNAME es algo similar al registro CNAME. Pero en lugar de proporcionar un alias para un solo nombre como CNAME, DNAME puede provee un alias para una parte completa en el árbol de un DNS: un dominio o subdominio.

Por ejemplo el registro DNAME se puede representar de la siguiente manera:

```
investigacion.ejemplo.com                IN  DNAME  r-and-d.ejemplo.com.
www.plasticos.r-and-d.ejemplo.com        IN  A      192.0.2.1
www.biotecnologia.r-and-d.ejemplo.com    IN  A      192.0.2.2
```

Con el registro DNAME trabajando, todos los registros y subdominios bajo r-and-d.ejemplo.com se encuentran también presentes bajo investigación.ejemplo.com. Al realizar la búsqueda www.biotecnologia.investigacion.ejemplo.com se puede obtener el mismo resultado al hacer la búsqueda www.biotecnologia.r-and-d.ejemplo.com. Con lo que se puede ser compatible con el servidor de nombre para ser resumido en un registro CNAME, sin embargo, al parecer viejas librerías de resolución no pueden manejar correctamente el registro DNAME. La idea detrás de bitlabels (algunas veces llamados etiquetas binarias, “binary labels”) es que el tradicional …4.3.2.1.in-addr.arpa o …e.f.f.3.ip6.int como mecanismo de delegación resulta poco útil porque sólo permite la delegación en 8 o 4 bits de frontera, respectivamente. Para conceptualizar un bitlabel se necesita expresar individualmente un nombre de dominio muy largo con la separación por periodos de bit, (En un sistema de nombre de dominio, los datos entre dos periodos son llamados “label”) sin embargo, dentro del protocolo DNS, un bitlabel es expresado por un simple fragmento de datos binarios, independiente del número de bits que contiene, en lugar de una larga lista de labels ASCII de cada uno de ellos en la zona de archivos de DNS. Bitlabel puede ser especificado en cualquier sistema binario, octal, hexadecimal o con un valor explicito, indicando la longitud en bits.
Por lo que se pueden mostrar varias representaciones de bitlabel de la misma información, por ejemplo:

```
\[xf0d2b496785a3c1e]   IN PTR www.example.com.
\[b1111000011010010101101001001011001111000010110100011110000011110] IN PTR
www.ejemplo.com.
\[o741512644547413217017 0/64] IN PTR www.ejemplo.com.
\[120.90.60.30].\[240.210.180.150] IN PTR www.ejemplo.com.
```

El primer bitlabel es hexadecimal, como lo describe la inicial x. El segundo es un binario (b) y el tercero es un octal (o). Porque un digito octal representa tres bits, la cadena de 22 caracteres, puede especificarse con 66 bits. El significado del /64, nos indica que sólo 64 bits pueden ser considerados parte de el bitlabel. Y para concluir, la última línea entre paréntesis cuadrados muestra la notación decimal. Ya que está notación es limitada a 32 bits, debe ser concatenada con dos bitlabels, y así completar los 64 bits. Se debe tener en cuenta que dentro de cada bitlabel, lo más natural es que el orden va del más significativo al menos significativo con la notación usada usualmente los labels menos significativos a más significativos del sistema de nombre de dominio regresan la forma de concatenación de los bitlabels. Son los 64 bits de la notación decimal que se encuentra en los paréntesis cuadrados la permitida, esa versión del bitlabel se puede ver en el ejemplo anterior, por lo que se puede expresar como \[240.210.180.150.120.90.60.30]. junto a DNAME y bitlabel se puede permitir el mapeo inverso de la información y los resultados pueden ser delegados como se muestra a continuación:

```
\[x20010DB81BFF/48].ip6.arpa.                    IN DNAME rev.ejemplo.com
\[xC001/16].rev.ejemplo.com                      IN DNAME srvrs.rev. ejemplo.com.
\[x0000000000000390/64].srvrs.rev.ejemplo.com.   IN PTR www.ejemplo.com.
www.ejemplo.com.                                 IN AAAA 2001:db8:1bff:c001::390
```

En la vida real, la primera línea puede ser asignada por un ISP, y estará en la zona de archivos en los ISPs, sin embargo, las otras líneas pueden estar en la misma zona de archivos o pueden ser extendidos a otras zonas para aumentar la flexibilidad y la facilidad de renumeración.

RFC 3595 AAAA y ip6.int.

En el IETF se genero un debate sobre los meritos y ventajas de los RFC 1886 [28] y el RFC2874 [29] y las formas en que hacían las cosas. Parte de está discusión se encuentra en los RFCs 3363 [32] y 3364 [33] que se realizó en el 2002. El argumento principal a favor del RFC 2874 [29] por su flexibilidad y soporte para las renumeraciones rápidas. El argumento en contra del uso del registro A6 para el almacenamiento de direcciones IPv6 en los DNS y el bitlabels para realizar el mapeo inverso, es que estos agregan una mayor complejidad y aumentan el tiempo que se necesita para realizar una búsqueda. En el RFC 3364 [33] también se encuentran las notas de cómo el registro A6 es optimizado para la lectura de información en los DNS. Es difícil de imaginar la forma en que la información en el DNS se mantendrá sincronizada con la dirección que en ese momento es usada por un equipo durante un proceso de renumeración.

Eventualmente, estos casos, condujeron a la conclusión de que los registros AAAA serían la mejor opción para almacenar las direcciones IPv6 en los DNS, y el método de nibble era la mejor forma para realizar el mapeo inverso. Un nibble consta de 4 bits, que se refiere a …1.0.0.2.ip6… técnica de mapeo inverso. Sin embargo, en el 2001 la Comisión de Arquitectura de Internet, IAB por sus siglas en ingles (Internet Architecture Board) publico en el RFC 3172 [34] declaró una preferencia hacia .arpa (ahora Direcciones y parámetro de área de ruteo), como una infraestructura de dominio de nivel superior, que resulto en una vuelta al RFC1886 [28] y ip6.int. Posteriormente, el uso de ip6.int fue obsoleto y se recurrió a ip6.arpa en la mejor práctica actual, mejor conocido como RFC 3152 [35]. Todas estas discusiones son concluidas en el RFC 3596 [24] en el 2003, con lo cual se estandariza el uso del registro AAAA y el método de nibble para la búsqueda inversa bajo ip6.arpa.

Los más importantes cambios hechos al RFC 1886 [28] fueron:

- Reemplazo del dominio IP6.ini por IP6.arp.
- Mención del tipo de consulta SRV en "Modificaciones a los tipos de consultas existentes".
- Se agregan consideraciones de seguridad.
- Se actualizaron referencias del RFC 1884 [26] al RFC 4291 [8].
- Se actualizaron referencias de trabajo en proceso del RFC 2893 [12].
- Se agregaron referencias al RFC 1886 [28], RFC 3152 [35], RFC 2535 [37] y RFC 2845 [38].
- Se actualizaron los documentos de resumen.
- Se agregaron tablas de contenidos.
- Se agregaron todos los derechos de autor.
- Se agrego una sección con consideraciones de IANA.
- Se agregaron declaraciones de la propiedad intelectual.

5.4. Herramientas para la manipulación de direcciones IPv6.

Como ya se ha mencionado el manejo de direcciones IPv6 es un cuanto complicado debido al tamaño de las mismas, debido a que se trata de un protocolo que se usa en capa tres, como las direcciones IPv4, se puede hacer uso de diferentes herramientas para dar solución a diferentes problemas al manejar las direcciones o querer obtener información de las mismas, se puede tener una ayuda para la correcta configuración de las redes y los host con la nueva versión del protocolo.

Los diferentes sistemas operativos que tienen soporte IPv6, cuentan con diferentes comandos que ayudan al manejo de direcciones IPv6, así monitorear y verificar su buen funcionamiento o configuración en las interfases de los hosts.

Hay diferentes distribuciones de **Linux** que se manejan actualmente dependiendo del tipo de aplicaciones que se deseen ejecutar, pero todas estas están basadas en el mismo kernel. El kernel de Linux que tiene soporte IPv6 es desde la versión 2.2.X.

Se pueden listar varias herramientas para el manejo de direcciones IPv6, en muchas de las ocasiones estas tiene que ser instaladas, ya que no todas las distribuciones contemplan su utilización.

IPv6calc.

Esta es una herramienta desarrollada para los sistemas operativos Linux, sirve para reconocer muchos tipos de formatos de direcciones IPv6, y dependiendo de la acción que se desee, será el formato de salida de la dirección, también puede mostrar información detallada de una dirección IPv6/IPv4.
Algunas de las utilidades que se le puede dar a este comando son:

La conversión simple de una dirección IPv6 a una consulta del PTR DNS:
./ipv6calc --out revnibbles.arpa 2001:db8::1
No input type specified, try autodetection...found type: ipv6addr
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.

También se puede ver el formato de entrada que fue autodetectado, al dar un ejemplo puede ser extendido, ipv6calc puede interpretar valor de máscara de red y el largo del prefijo.
./ipv6calc -q --out revnibbles.arpa 2001:db8::1/64
0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.

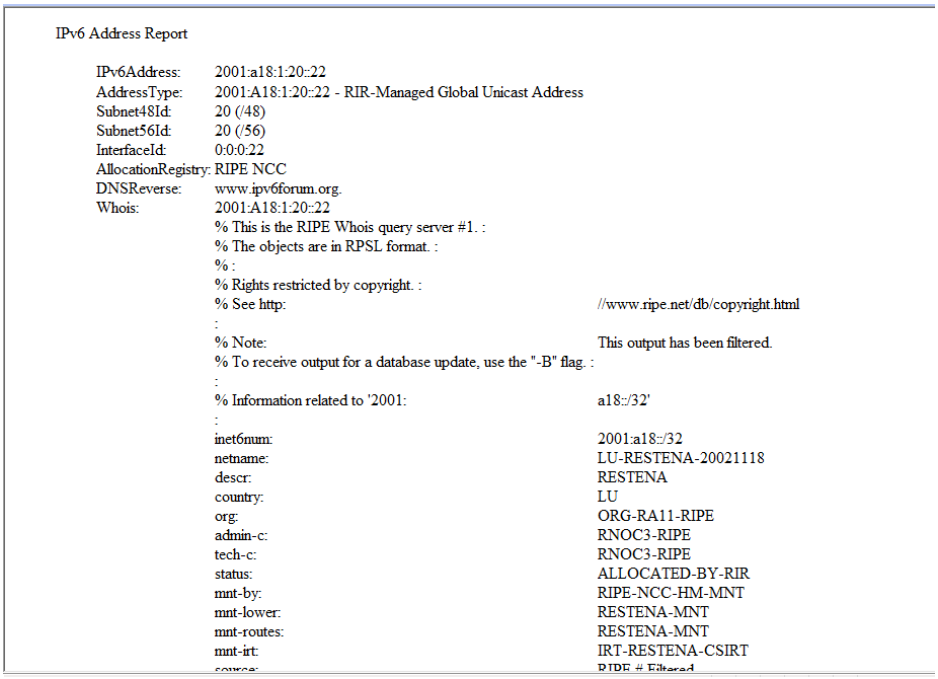
Este comando también puede ayudar a interpretar direcciones IPv4 que se encuentran en los túneles 6to4.
./ipv6calc -q --action conv6to4 --in ipv6 2002:c0a8:fb61::1 --out ipv4
192.168.251.97

Sipcalc

Es una consola avanzada para realizar cálculos sobre las direcciones IPv6/IPv4 de una subred. Ya que puede tomar múltiples formatos de entrada como direcciones IPv4, direcciones IPv6,interfases o hostnames y con éstos realizar operaciones para dar como resultado una gran variedad de información acerca de la subred. Para las direcciones IPv6 se pueden simplificar o expandir según sea la entrada que se proporcione, generar direcciones inversas para DNS, calcular los prefijos IPv6 para redes entre otras opciones.

También se encuentran otras herramientas disponibles para el manejo de direcciones IPv6, ya que vía web se puede obtener información de los prefijos y direcciones IPv6, a través de la pagina http://www.potaroo.net/cgi-bin/ipv6addr. Por lo que si se desea saber si una la dirección tiene un DNS reservado, su ubicación y más información acerca de esa dirección, se presenta un resumen de la misma con todos los datos que pertenecen a ese prefijo.

Ejemplo:
Al acceder a la página y pedir la consulta de la dirección 2001:a18:1:20::22 se muestra la figura 5.4



IPv6 Address Report

IPv6Address:	2001:a18:1:20::22
AddressType:	2001:A18:1:20::22 - RIR-Managed Global Unicast Address
Subnet48Id:	20 (/48)
Subnet56Id:	20 (/56)
InterfaceId:	0:0:0:22
AllocationRegistry:	RIPE NCC
DNSReverse:	www.ipv6forum.org.
Whois:	2001:A18:1:20::22
	% This is the RIPE Whois query server #1. :
	% The objects are in RPSL format. :
	% :
	% Rights restricted by copyright. :
	% See http: //www.ripe.net/db/copyright.html
	:
	% Note: This output has been filtered.
	% To receive output for a database update, use the "-B" flag. :
	:
	% Information related to '2001: a18::/32'
	:
	inet6num: 2001:a18::/32
	netname: LU-RESTENA-20021118
	descr: RESTENA
	country: LU
	org: ORG-RA11-RIPE
	admin-c: RNOC3-RIPE
	tech-c: RNOC3-RIPE
	status: ALLOCATED-BY-RIR
	mnt-by: RIPE-NCC-HM-MNT
	mnt-lower: RESTENA-MNT
	mnt-routes: RESTENA-MNT
	mnt-irt: IRT-RESTENA-CSIRT
	source: RIPE # Filtered

Figura 5.4 Resultado de la consulta de la dirección 2001:a18:1:20::22 en http://www.potaroo.net.

NDisc6
Es un kit de herramientas de diagnostico IPv6 para Linux y BSD, que reúne una pequeña colección de programas que proporcionan información para redes IPv6. Los programas incluidos son:

- ndisc6, herramienta de descubrimiento de vecino ICMPv6.
  Realiza el descubrimiento de vecino IPv6 en la red, ya que este mecanismo sustituye ARP de IPv4.
- rdisc6, herramienta de descubrimiento de ruteadores.
  Realiza preguntas a los ruteadores IPv6 de la red sobre los prefijos anunciados en la misma. También puede ser usado para forzar a los host locales a que puedan realizar la autoconfiguración de su dirección IPv6.
- Tcptraceroute6, IPv6 traceroute, usa el paquete TCP/IPv6 para realizar una ruta a una dirección IPv6.
- Traceroute6: IPv6 traceroute, usa el paquete UDP/IPv6 para realizar una ruta a una dirección IPv6.

ipv6gen, generador de prefijos IPv6.
Es una herramienta que genera una lista de prefijos IPv6 de longitud dada, cumpliendo con lo especificado en el RFC 3531.
Este programa es de gran ayuda para poder construir esquemas de direcciones IPv6 o asignar prefijos automáticamente. Ipv6gen está estructurado dentro de funciones que pueden ser usadas en programas propietarios.

Un ejemplo del uso de este programa se muestra a continuación en la tabla 5.5

```
[~/IPv6/ipv6gen]> ./ipv6gen.pl 2001:1508:1003::/48 64
2001:1508:1003:0000::/64
2001:1508:1003:0001::/64
2001:1508:1003:0002::/64
2001:1508:1003:0003::/64
...

2001:1508:1003:FFFA::/64
2001:1508:1003:FFFB::/64
2001:1508:1003:FFFC::/64
2001:1508:1003:FFFD::/64
2001:1508:1003:FFFE::/64
2001:1508:1003:FFFF::/64
```

Tabla 5.5 Herramienta generadora de prefijos para una subred.

Otras herramientas que se pueden utilizar para obtener información sobre los servidores, host o direcciones IPv6 de una red, se cuenta con los diferentes comandos de los sistemas operativos Windows y Linux que cuentan con soporte IPv6, como es el caso de ping en los diferentes sistemas operativos que acepta direcciones IPv6 Unicast de enlace local como globales.

Cuando se hace un ping a una dirección de enlace local, es necesario especificar la interfaz de red con la cual trabaja el host que enviara los mensajes, de lo contrario no se podrá realizar la comunicación entre los equipos. La notación para especificar la interfaz es [dirección IPv6]%[Número Interfaz].

Traceroute6 y tracert del sistema operativo Linux y Windows respectivamente, son herramientas inherentes de ambos sistemas, con la cual nos describe el camino por el cual viajan los paquetes de un origen hasta un destino.

Nslookup es una herramienta que nos muestra información de un host o dominio a partir de su nombre, debido a que este comando realiza consultas a los DNS para así obtener la dirección IPv4/IPv6 del sitio, su dirección inversa y más información sobre el tipo de registro con el cual trabajan los servidores.

Capítulo 6
IPv6 en la UNAM

6.1. Introducción

En la Universidad Nacional Autónoma de México, la interacción con nuevas tecnologías es primordial para el avance del conocimiento, mantenerse a la vanguardia tecnológica y estar a la altura de las mejores instituciones educativas, es por eso que su colaboración en la aplicación de nuevos protocolos de Internet es de gran importancia, dado que, las exigencias tecnológicas de nuestros días deben ser cubiertas para prever un atraso tecnológico y académico.

Como se ha mencionado en capítulos anteriores, debido a la creciente demanda de nuevas redes, de mayor espacio de direccionamiento y a un mejor protocolo para las nuevas tecnologías de Internet, el IETF convocó a un proyecto de Internet de siguiente generación, IPng que también se le conoce como IPv6. Para que se implementaran nuevas características a una nueva versión del protocolo de Internet; entre las cuales se puede mencionar, que su espacio de direcciones es inmensamente grande, la capacidad de que los equipos puedan realizar autoconfiguración, mejor soporte a la calidad de servicio, seguridad, etc.

Así, después de definido y aceptado IPv6, diversas organizaciones comenzaron con una transición gradual de IPv4 a IPv6.
Uno de los proyectos internacionales más importante fue el 6Bone, que aplicó los conceptos y puso en marcha el uso de IPv6 en una red virtual, a la cual se conectaban islas IPv6 a través de túneles, que son conexiones punto a punto entre dos nodos. En este proyecto participaron 55 países entre ellos México.

En nuestros días, existen organizaciones mundiales, grupos de trabajo y capítulos regionales del foro IPv6, que son formados por proveedores líderes en soluciones de telecomunicaciones, proveedores de Internet, así como investigadores e instituciones educativas que trabajan para implementar realizar investigaciones, y con ello aportar experiencias para tener una transición más sencilla a la nueva versión del protocolo de Internet.

La UNAM coordina un grupo de trabajo IPv6, el cual se encuentra dado de alta en la Corporación Universitaria para el Desarrollo de Internet, CUDI por sus siglas; la cual, tiene como objetivo desarrollar conocimiento, coordinar y promover acciones para el desarrollo de soluciones para las redes de cómputo y comunicaciones actuales, enfocadas al crecimiento del desarrollo científico y tecnológico.

6.2. Historia

Las comunicaciones telefónicas y de datos en la UNAM tomaron un gran auge a finales de 1960 y principios del año 1970. En este período se realizaron las primeras conexiones de teletipos hacia una computadora central, utilizando líneas telefónicas de cobre, de la recién instalada red telefónica dentro de la universidad.

Debido a lo anterior se inició en la UNAM una serie de pruebas que usaban la tecnología para el beneficio de los universitarios así como de las instituciones que no se encontraban dentro de ésta, ya que se realizaban una gran cantidad de conexiones de terminales de caracteres, graficación e impresión hasta la interconexión de estaciones de trabajo remotas; todas estas conexiones se realizaron a través de líneas telefónicas.

A partir de la segunda década de los años 80s, surge en la universidad el interés de hacer un cambio en las comunicaciones. Así en 1987, la universidad establece la primera conexión a la Red Académica BITNET, mediante enlaces telefónicos, desde Ciudad Universitaria hasta el Instituto Tecnológico de Estudios Superiores de Monterrey (ITESM), y de ahí hasta San Antonio, Texas en los Estados Unidos de América. Por lo que la UNAM buscó consolidar su enlace a esa red internacional mediante la computadora IBM 4381, la cual sirvió como residencia de correo electrónico y otros servicios de BITNET. Dentro de ese proceso se inició la conexión de terminales IBM con emulación 3270, estableciéndose además, un enlace con la Red TELEPAC de la Secretaria de Comunicaciones y Transportes, bajo la finalidad, nunca lograda, de brindar este servicio a nivel nacional.

Fue hasta 1989, cuando la UNAM establece un convenio de enlace a la red de la Fundación Nacional de Ciencia, NSF por sus siglas en inglés(National Science Foundation) en EUA a través de su instituto de astronomía, haciendo uso del satélite mexicano Morelos II entre el instituto de la UNAM y el UCAR-NCAR con residencia en Boulder Colorado. Con este acto se inició una gran actividad en las comunicaciones dentro de la UNAM, así como la adquisición de computadoras personales y su interconexión e intercomunicación en redes de área local, principalmente en las dependencias del Subsistema de la Investigación Científica, lo cual dió paso al desarrollo de la infraestructura de las comunicaciones con fibra óptica, y a establecer más enlaces satelitales hacia diferentes estados de la república.

En 1990, la UNAM fue la primer institución en Latinoamérica que se incorporó a la red de redes mundial Internet, que enlaza a millones de equipos y decenas de millones de usuarios en todo el mundo.

La UNAM siempre se ha caracterizado por trabajar a favor de la investigación, impulsar el conocimiento y la participación académica con instituciones que ayuden al desarrollo tecnológico. Dado que el IETF había recomendado el uso de IPng en el RFC 1752 [2], y después de la aprobación de diferentes organismos para este protocolo, la UNAM se dio a la tarea de encaminarse al uso de la nueva versión del protocolo y participar en el desarrollo de proyectos, investigaciones, pruebas, difusión e instalación de IPv6.

Para finales de 1998 se constituye el proyecto IPv6 de la UNAM; así por los primeros trabajos realizados y debido al liderazgo demostrado, refrendó ser una institución que siempre está a la vanguardia en el ámbito nacional. Logró en menos de un año, para junio de 1999, ser el primer nodo de la red 6Bone en México, debido a que su programa de pruebas y trabajos después contempló implementaciones de pila dual IPv4/IPv6, túneles, software de conexión, aplicaciones multimedia, servidores Web, DNS, autoconfiguración, calidad de servicio, IPv6 sobre ATM, conexión con otras redes internacionales de IPv6 (como 6REN), IPv6 en Internet2, etc.

Posteriormente ese mismo año, en el mes de septiembre, la UNAM fue aceptada como uno de los 68 nodos del backbone que operaban en ese momento en 6Bone como se observa en la figura 6.1, gracias a que siempre ha presentado un nivel académico sobresaliente y a los resultados de las pruebas que se realizaban en la institución, con esta aceptación, se le asigna un rango de direcciones tipo pTLA(Top-Lever Aggregation):3ffe:8070::/28. Con este hecho la UNAM se convirtió en el primer nodo de este tipo en México y el tercero en Latinoamérica. Por lo que al continuar con los trabajos y pruebas en la red universitaria, para octubre de 2000 se obtuvo un bloque de direcciones tipo sTLA 2001:0448::/35, para poder ofrecer a los usuarios un servicio de producción con IPv6. De tal forma que la universidad ha podido tener la capacidad de poder delegar direcciones y configurar túneles a instituciones en México y a cualquier institución interesada en realizar pruebas con IPv6.

Figura 6.1 Conexión de RedUNAM y diversas universidades al 6bone

Hoy en día se trabaja con instituciones de México, para que en su conjunto se puedan hacer pruebas de aplicaciones IPv6, una vez que se ha realizado una conexión IPv6 entre las instituciones participantes. Algunas de las instituciones mexicanas que han puesto interés en el desarrollo y pruebas IPv6, son: Universidad Autónoma de Chiapas, Universidad Autónoma de Guerrero, Universidad Autónoma del Estado de Hidalgo, Universidad Autónoma de Nuevo León, Universidad de Guadalajara, Universidad la Salle, Universidad de Colima, Instituto Politécnico Nacional, Instituto Tecnológico de Estudios Superiores de Monterrey, Instituto Tecnológico de Oaxaca, Instituto Tecnológico de Mérida, Instituto Tecnológico Autónomo de México, la Corporación Universitaria para el Desarrollo de Internet (CUDI), LANIA, CICESE, PEMEX, ASTER, ISOC-MEX, etc.

Así mismo, la colaboración de instituciones de Latinoamérica es de gran importancia, ya que casi siempre están dispuestas a participar y compartir el conocimiento generado, y con esto aumenta la difusión de cada proyecto IPv6 de la región a la que pertenecen; por lo que se pueden nombrar al Instituto de Informática de la Universidad Austral de Chile (UACH), Compendium de Argentina, Laboratorio de Investigación en Nuevas Tecnologías de la Universidad Nacional de La Plata (LINTI-UNLP) de Argentina, EAFIT de Colombia, entre otras.

Para abril del 2000 uno de los logros del proyecto de IPv6 de la UNAM fue que se volvió miembro del IPv6 Forum, gracias a sus trabajos realizados dentro del mismo, lo cual colocó a nuestra universidad como una de las precursoras de la nueva versión del protocolo en nuestro país, y así para septiembre del mismo año fue el comienzo del capítulo México del IPv6 Forum, que tiene como objetivo, incrementar la investigación, desarrollo, difusión y utilización de IPv6 en México y Latinoamérica.

6.3. Servicios

RedUNAM es una de las redes educativas de telecomunicaciones digitales más avanzadas y de gran tamaño de Latinoamérica, por lo que se considera uno de los centros de tráfico de información más importantes del país, cumpliendo con la tarea estratégica de intercomunicar a la comunidad universitaria entre sí, y a su vez comunicando al resto de las comunidades académicas y científicas del mundo.

En nuestros días es de gran importancia la participación de la comunidad universitaria para mantener en óptimo estado las redes de la institución y así dotarla de una moderna infraestructura de telecomunicaciones y cómputo; dar un valor al método de enseñanza-aprendizaje para compartir las investigaciones que se realizan y proponer un camino para las próximas a realizar, proporcionando a su personal docente y de investigación de todas las herramientas de la tecnología informática para el desarrollo de sus actividades, integrar a los alumnos de todos los niveles de educación a la cultura informática y difundir al público en general, los conocimientos y la tecnología que se generan en la UNAM, resultado de investigaciones en las múltiples áreas del conocimiento.

Debido a la gran cantidad de usuarios, a los avances tecnológicos y a las necesidades que requieren los docentes e investigadores para el desarrollo de sus proyectos, RedUNAM brinda una gran variedad de servicios que impulsan la formación de conocimiento de la comunidad universitaria.
Los servicios pueden dividirse de la siguiente manera:

Aquellos que pueden hacer uso los usuarios, los cuales son:

- Asesoría y soporte técnico a equipo de videoconferencia a usuarios de la UNAM, y a externos.
- Claves personalizadas para extensiones de la UNAM larga distancia y celular.
- Facturación telefónica UNAM.
- Conexión a la red de la UNAM.
- Red de emergencia UNAM.

- Correo electrónico: servidor.unam.mx
- Cambio de password en servidor.unam.mx
- Llamadas a celular o larga distancia de extensiones de la UNAM.
- Asignación gratuita de acceso a Internet vía módem para académicos, funcionarios e instituciones en convenio.
- Ayuda para configurar el acceso a Internet (módem).
- Enlace satelital (datos).
- Enlace satelital (línea telefónica).
- Conectividad a Internet de una PC.
- Conectividad de una red local a Internet.
- Líneas directas de Telmex.
- Servicios digitales para líneas directas de Telmex.
- Cambio de claves personalizadas (códigos lada) para líneas directas de Telmex.
- Cambio de domicilio de líneas directas de Telmex.
- Llamadas a celular de líneas directas de Telmex.
- Claves personalizadas (códigos lada) para líneas directas de Telmex.
- Solicitud de teléfonos para casetas públicas móviles de Telmex y Telcel para eventos.
- Instalación de líneas directas y de extensiones de Telmex.
- Eventos que utilicen las líneas ISDN.
- Tratamiento de incidentes y quejas de seguridad.
- Servicio telefónico.
- Instalación de cableado telefónico interno y externo.
- Teléfono digital.
- Teléfono secretarial.
- Teléfonos unilínea.
- Servicios de datos/voz/videoconferencia.
- Acceso a Internet vía módem.
- Asesoría en la instalación de enlaces para la conexión de aulas de videoconferencia.
- Asesoría técnica de cuentas de correo.
- Asignación, actualización y seguimiento de IPv6.
- Correo de los sitios que tienen Web Hosting.
- Actualización de datos.
- Avisos etc.

Por lo que, los servicios anteriores a su vez se encuentran soportados por otros servicios que permiten la interacción y comunicación entre diversos host. Estos servicios están administrados de la siguiente manera:

- Centro de Información de RedUNAM, NIC-UNAM.
  Es el que se encarga de distribuir la información de los servicios de red, soportarlos dentro de la RedUNAM y dar la capacitación necesaria a los usuarios sobre estos servicios.
  Los servicios que administra el centro de información de RedUNAM son:
  - Asignación de direcciones IP, tanto de IPv4 como de IPv6, lo que autoriza a los responsables de una dependencia o institución conectada a la UNAM el uso de un rango de direcciones IP pertenecientes o asignadas a la UNAM.

  - Asignación de dominios, lo que autoriza a los responsables de una dependencia la utilización de un dominio con el objetivo de asociarlo a sus direcciones IP, así como la posibilidad de enviar solicitudes de Altas y/o Bajas de nombres o alias. Cualquier solicitud debe ser realizada mediante los formatos y procedimientos establecidos por NIC-UNAM.
  - Servicio de nombres, con esto se dan las altas y/o bajas en los DNS de la UNAM, ya que es conveniente para los equipos que tienen una dirección IP, tener un nombre y un dominio asignado, para permitir el envío de correos electrónicos, acceso a paginas web, etc.; estas operaciones sólo pueden ser realizadas por los administradores de NIC-UNAM.
- Centro de Monitoreo RedUNAM, NOC-UNAM.
  Es responsable del monitoreo del estado de la Red y la atención, resolución y análisis de incidentes y problemas que afecten en la disponibilidad y funcionalidad de la infraestructura de red. En otras palabras, es el encargado de mantener funcionando de manera eficiente la interconexión de las redes locales, los enlaces de área amplia y el backbone de la red universitaria.

6.4. Situación actual del soporte IPv6 en RedUNAM.

En RedUNAM, se han hecho varias modificaciones desde que se delegó su espacio de direcciones por el 6Bone, por lo que, al hacer pruebas de conectividad con otras instituciones, delegarles espacios de direcciones y la posibilidad de que pudieran tener conexiones IPv6 por túneles, se ha podido implementar pruebas como pilas dual IPv4/IPv6, servidores Web, DNS, aplicaciones multimedia, autoconfiguración, calidad de servicio, IPv6 nativo, etc. Para que todos estos servicios y pruebas se llevaran acabo el soporte IPv6 en un segmento de RedUNAM se implementó de manera satisfactoria, constituido por clientes bajo diferentes sistemas operativos, ruteadores de diversos fabricantes y conexiones a 6Bone. En la figura 6.2 se pueden ver algunos de los nodos conectado a RedUNAM cuando existía 6Bone.

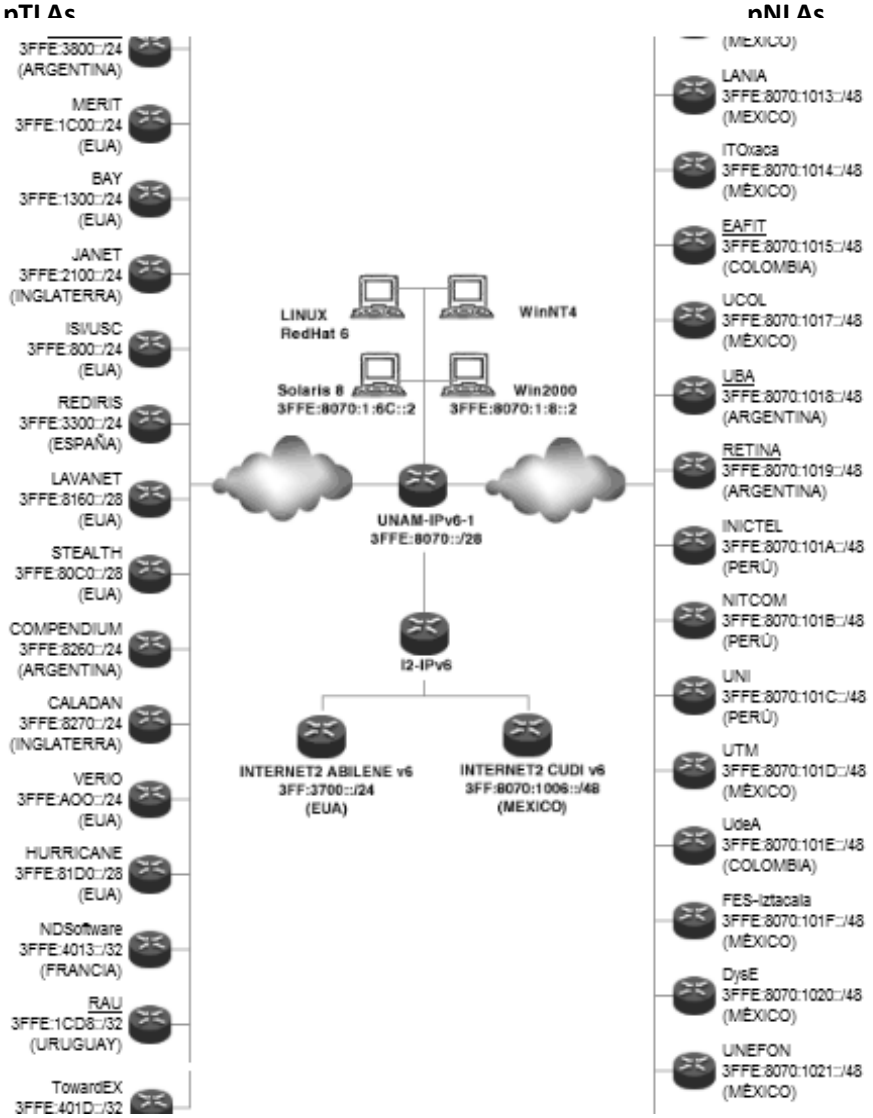


Figura 6.2 Conexiones y asignaciones en RedUNAM con soporte IPv6 en proyecto 6bone.

Así la UNAM que en principio fue el primer nodo IPv6 de su tipo en México, y que en su primera etapa estuvo funcionando como una red de pruebas, posteriormente se convirtió en una red de producción, en la cual conviven diferentes sistemas operativos, equipos y aplicaciones que soportaban IPv6 por defecto o se han podido adaptar por medio de parches que se encuentran disponibles en Internet, con los que se llevaron a cabo pruebas que incluyen temas como:

- Pila dual IPv4/IPv6 para equipos con diferentes plataformas de Windows, Mac y Unix.
- Túneles configurados IPv6 sobre IPv4, IPv6 en IPv4.
- Método de conexión por túnel 6to4.
- Servidores Web para sistemas operativos Solaris, Linux y Windows.
- Autoconfiguración con ruteradores de diferentes fabricantes, así como la autoconfiguración con diferentes sistemas operativos y métodos de transición.

- Software traductor IPv4/IPv6 para Windows.
- Análisis de tráfico IPv6.
- Desempeño IPv6 vs IPv4.

Estas son algunas de las pruebas que se han realizado, con fines de ampliar el conocimiento y dar difusión a la nueva versión del protocolo, debido a que el proyecto de 6Bone terminó y gracias a que la UNAM recibió un bloque de direcciones para producción se cambió la estructura de la red, adaptándose a las nuevas conexiones y nuevas direcciones delegadas. Como se muestra en la figura 6.2

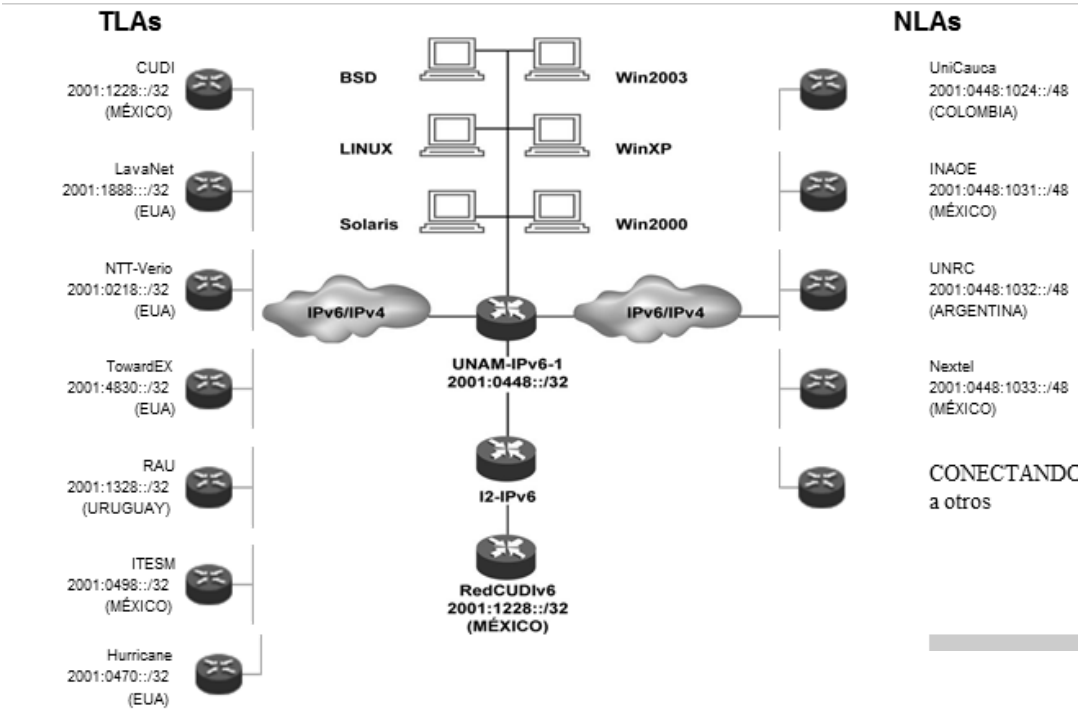


Figura 6.1 RedUNAM al término de 6Bone.

6.5. Actividades de difusión e información sobre IPv6.

Las actividades de difusión e información de las investigaciones y pruebas sobre IPv6 son realizadas por la dirección de telecomunicaciones de la UNAM en el laboratorio de tecnologías emergentes de redes (NETLab).

Debido a que las diferentes pruebas que lleva acabo el laboratorio se realizan con diversos equipos de cómputo, sistemas operativos y aplicaciones de diversos fabricantes, no se busca ser una entidad especializada en una plataforma en especifico, sino como un generador de nuevos conocimientos, soluciones óptimas a las necesidades actuales en cuanto a tecnología y un difusor de información basados en los resultados de dichas pruebas, para que así la comunidad universitaria e instituciones sean apoyadas en sus proyectos.

La UNAM como generadora de conocimientos es de las más importantes en Hispanoamérica, debido al nivel de sus investigaciones y difusión de las mismas, por lo que en nuestros días es primordial llevar el conocimiento tanto a la comunidad universitaria interna como de otras instituciones y cualquier persona que se encuentre interesada. Como se había mencionado con anterioridad RedUNAM presta el servicio Web, el cual es de suma importancia para las difusión de diferentes materiales de ámbito informativo, investigación, estado de pruebas y eventos en los cuales se tiene participación, y con ello poder ofrecer esta información de manera sencilla a cualquier persona interesada.

La información en los servicios Web abarca gran cantidad de temas y es una de las más consultadas por ser de fácil acceso para los administradores e investigadores; por ejemplo, los protocolos de Internet. El laboratorio cuenta con la página www.netlab.unam.mx para dar de forma concreta la información sobre los diferentes proyectos que se realizan para la innovación en el ámbito tecnológico, por lo que se ofrece una serie de documentos teóricos para que se pueda tener el conocimiento básico sobre IPv6 y pruebas que se llevan acabo, así como tutoriales para las buenas prácticas sobre la utilización de las tecnologías de redes. Al ofrecer esta información se contribuye al desarrollo académico en la universidad, y se ofrece la oportunidad de tener un acercamiento a los proyectos que se realizan en el laboratorio, así poder atraer patrocinios y recursos humanos que deseen participar en las pruebas, y con ello capacitar al estudiantado y personal interesado en el manejo de diferentes tecnologías.

Otras actividades de difusión que realiza la dirección de telecomunicaciones es la impartición de talleres, pláticas y conferencias, haciendo uso de las instalaciones de la Dirección General de Servicios de Cómputo Académico (DGSCA), como el auditorio o las salas de videoconferencias con transmisiones a nivel nacional y varios países de habla hispana, para poner al tanto sobre el desarrollo de las investigaciones y fomentar los lazos entre universidades.

Así los miembros que se ven involucrados en los diferentes proyectos tienen la oportunidad de difundir el conocimiento adquirido y de tener una participación con las comunidades académicas, no sólo de la UNAM, sino de diferentes universidades del continente.

6.6. Actividades actuales y pendientes.

Actualmente ya se tienen propuestas para lo que podría ser el direccionamiento IPv6 para RedUNAM principalmente en la parte de datos, sólo faltaría mejorar el correspondiente para la red de voz. Estas propuestas se hicieron en base a la topología actual de la red de datos en donde están considerados los cuatro nodos principales del Backbone los cuales son: Arquitectura, IIMAS, Zona Cultural y DGSCA, una parte de distribución y otra para el acceso. De la misma forma se contempla el direccionamiento para VLANs, direcciones de Loopback y los prefijos de asignación a las dependencias para sus redes LAN; todo lo anterior se realizó con el prefijo IPv6 2001:1218::/32 delegado a la UNAM por parte de LACNIC.

De ser necesario, haría falta una nueva revisión y actualización de las políticas para la asignación de prefijos IPv6 y nombres de dominio. Se tiene también una propuesta de políticas elaborada en conjunto con NIC-UNAM, que actualmente se está revisando para agregar o redefinir algunos puntos.

Por lo que algunas de las actividades adicionales que convendría terminar de realizar son:

- Elaborar un listado minucioso de actualizaciones de Hardware y Software necesarias en los equipos de ruteo y contar con un documento en donde se describan los equipos y versiones de IOS necesarios para que Red UNAM soporte IPv6 e IPv4 de forma nativa o con mecanismos de transición.
- Elaborar también un listado de actualizaciones de Hardware y Software para los servidores principales en uso y contar con un documento en donde se describan los equipos y versiones de aplicaciones necesarios para que RedUNAM soporte IPv6, sin dejar de soportar IPv4 en los servidores.
- Implementación de relays locales, los cuales sean un servicio propio de la UNAM y que no se necesite de relays externos para ofrecer conectividad IPv6 mediante este mecanismo.
- Actualización permanente a guías de buenas prácticas de configuración de IPv6 para los diferentes sistemas operativos.
- Implementación de uno o varios Servidores de Nombres de Dominio que transporten nativamente IPv6 en RedUNAM.

Capítulo 7
Planeación de una transición IPv4 a IPv6 en RedUNAM

7.1. Aspectos a considerar para ofrecer servicios y aplicaciones con soporte IPv6.

El soporte de IPv6 está disponible para la mayoría de las plataformas actuales tanto en equipo de cómputo como para ruteadores; ya sea en un nodo sencillo, una pequeña red o en una gran red empresarial.
Si se desea tener comunicación con otros sistemas IPv6 remotos, es vital obtenerla con la Internet global IPv6, configurando los sistemas locales.

La puesta en marcha de redes que sólo utilizan IPv6, pueden ser los primeros pasos para experimentar con esta versión ya que la realidad práctica muestra que sitios que despliegan IPv6 no realizan la migración directamente a la nueva versión del protocolo, sino primero hacen una transición a un estado intermedio donde conviven IPv4 e IPv6 utilizando el mecanismo de pila dual.

La expansión de la funcionalidad de IPv6 desde una pequeña a una grande red puede ser un proceso complejo y difícil, pero si se planea eficazmente, el despliegue puede hacerse en una manera escalonada y controlable. Para un sitio grande hay diferentes requisitos y condiciones, lo que hace necesario emplear varios mecanismos de transición según las características de la infraestructura así como de las tecnologías que se utilizarán en convivencia con IPv6, por ejemplo, una subred dada, ambiente inalámbrico o móvil, una tecnología dial-up, etc.

La nueva versión del protocolo no sólo es dotar de más direcciones IP a la Internet sino que un cambio de IPv4 a IPv6 refleja una ventaja a nivel tecnológico importante, con lo cual se debe de tener conocimiento de los nuevos conceptos, que no es una tarea que se da de un día a otro. La anticipación y el conocimiento de los equipos con la última tecnología son clave para ofrecer servicios y aplicaciones IPv6 a todos los que lo necesiten.

Un aspecto importante para que las aplicaciones soporten IPv6 es que éstas dependen directamente del código fuente el cual debe estar listo para utilizar la nueva versión del protocolo, ya que anteriormente cuando se programaba una aplicación que trabajaba entre hosts, se pensaba en un solo formato de dirección y cierta cantidad de memoria para la misma, por lo general se trataba de una dirección IPv4, que se puede representar como “a.b.c.d” y no se tomaba en cuenta otra versión IP, como una dirección IPv6 que se representa como “X:X:X:X:X:X:X:X”, lo que significa un cambio considerable en los analizadores de direcciones ya que para las direcciones IPv4 se usa el punto “.” como separador mientras que en IPv6 se usan los dos puntos “:”, lo que puede provocar ambigüedad en las URLs, es por eso que deben ser revisados para adecuarlos al nuevo formato de la dirección. En el caso concreto de las API de socket para la capa de transporte existen los siguientes puntos a considerar:

- Estructura de datos para las direcciones IP, dados por sockaddr_in, sockaddr_in6 y sockaddr_storeage.
- Funciones del API de comunicaciones, socket(), bind(), connect(), read()/write()…
- Funciones de conversión de direcciones entre el formato de presentación y las estructuras de datos de direcciones.
- Opciones de configuración de red.

Por lo que para el avance en el desarrollo de aplicaciones con soporte de IPv4 e IPv6 es de gran importancia que las nuevas aplicaciones deban ser independientes de la versión del IP. La selección de la estructura de dirección de socket que se utilizan en la programación de las aplicaciones se muestra en la figura 7.1

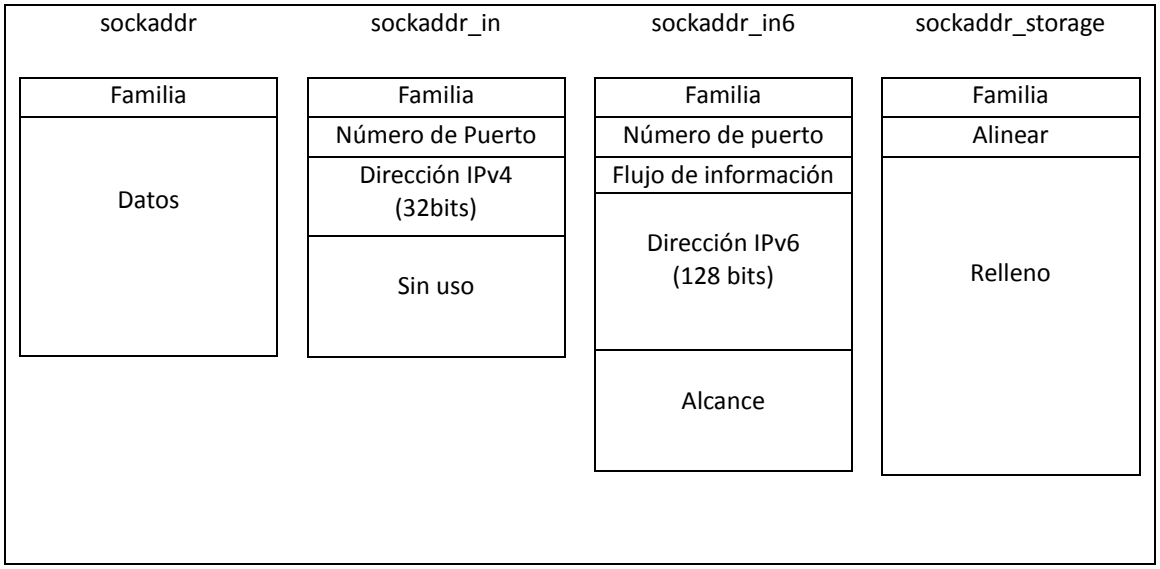


Figura 7.1 Estructura de los datos de sockaddr

Otros puntos a tomar en cuenta son:

- La selección de la dirección IP. Este trabajo radica en los hosts, en que puedan resolver automáticamente la selección de la dirección origen y destino, siguiendo una serie de reglas definidas en el RFC 3484 [36].
- Fragmentación a nivel de aplicación. Es el cálculo del tamaño del fragmento para que no haya degradación de prestaciones por fragmentación.
- Almacenamiento de direcciones IP. No almacenar direcciones IP ya que pueden cambiar. Si es necesario almacenar sólo nombres y solicitar la resolución en el momento se que se necesite.

Para que se empiecen a dar estas modificaciones es necesario hacer una revisión exhaustiva del código fuente de las aplicaciones, con ayuda de herramientas disponibles, para que se puedan tomar en cuenta todas las consideraciones y se tengan aplicaciones que soporten no sólo IPv6 sino también IPv4 mientras coexistan ambas versiones.

Finalmente para tener un panorama de los aspectos a considerar al ofrecer conectividad IPv6 se pueden mencionar las etapas que se deberían dar para realizar la transición:

1- Idear un plan de asignación de direcciones IPv6.
2- Obtener un espacio de direcciones IPv6, del NIC nacional o RIRregional, que puede ser un/48 sTLA, o uno de menor tamaño, de acuerdo a los requerimientos de las políticas cumplidas.
3- Estudiar las herramientas disponibles para el manejo y monitoreo de las redes.
4- Seleccionar el método de transición adecuado para el transporte sobre la infraestructura de red.
5- Seleccionar el protocolo de ruteo IPv6 adecuado y diseñar políticas de ruteo.
6- Implementar el método de transición seleccionado que ayude a la transición.
7- Implementar cualquier servicio que facilite la convivencia con IPv6.
8- Seguir las mejores prácticas para un despliegue seguro del método de transición.
9- Seguir los pasos 2 al 8 para cualquier red regional.
10- Habilitar IPv6 en los equipos de la red o dominio donde se desea implementar.

7.2. Escenarios posibles de convivencia de IPv4 e IPv6

En la actualidad todos los registros regionales (RIRs) han informado que las direcciones IPv4 están en un proceso de agotamiento y han estimado fechas para las cuales las direcciones se considerarán como agotadas, por lo que la asignación de los últimos bloques de direcciones y la administración adecuada de las asignadas será un trabajo con el cual se tratará de alargar lo más posible la disponibilidad de IPv4. La coexistencia de ambas versiones es un escenario necesario, que actualmente es una realidad, y que se está dando día a día en las grandes y pequeñas redes, con la conexión de islas IPv6, creación de túneles a diferentes dominios y proponiendo la mejor alternativa para seguir impulsando el avance de las redes con la nueva versión del protocolo.

Es por esta razón que los mecanismos de transición que se han mencionado, cobran gran importancia, ya que no sería posible imaginar la transición a la nueva versión del protocolo sin la ayuda de los mismos, así que es primordial que IPv4 e IPv6 coexistan por mucho tiempo en las redes de datos actuales y futuras.

Así como se proponen mecanismos para realizar conexiones entre redes, se deben de proponer actividades de formación específicas en las universidades y empresas a fin de difundir las políticas de implementación y las buenas prácticas con los diferentes métodos de transición.

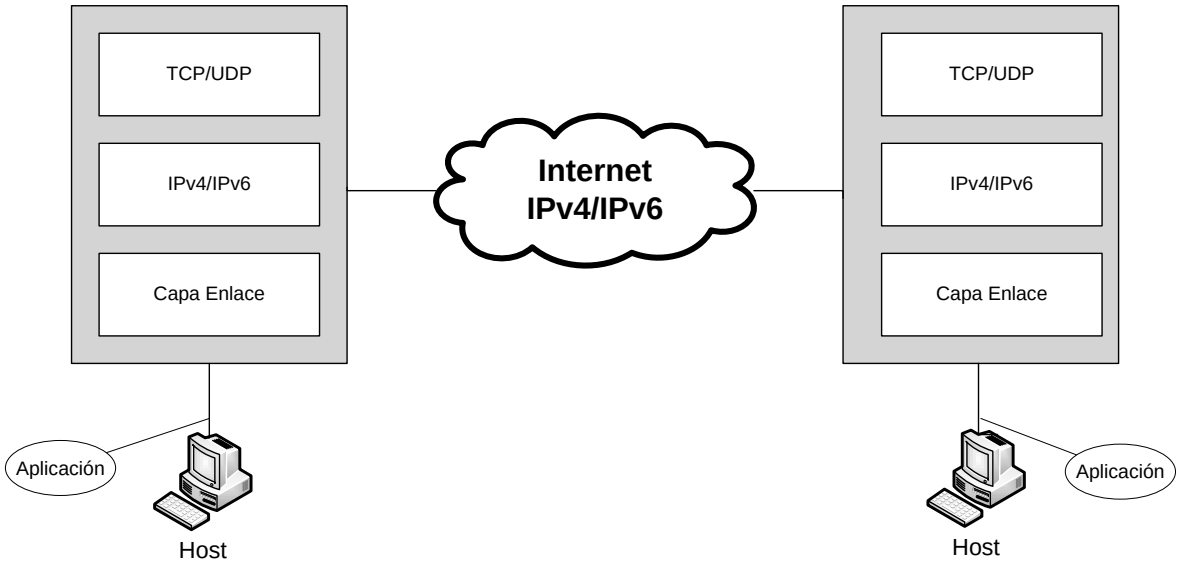


Figura 7.2 Arquitectura de la transición de IPv4 a IPv6.

Como se muestra en la figura 7.2 lo que se puede hacer en el proceso de transición de IPv4 a IPv6 es definir un grupo de hosts finales o redes, las cuales pueden encontrarse conectando a diferentes redes, contemplando:

- Redes sólo IPv4.
- Redes sólo IPv6.
- Redes Duales.

Una parte importante en el proceso de transición se lleva a cabo en la pila TCP/IP, ya que es fundamental definir por cuál versión de protocolo se llevará acabo la comunicación, por eso el mecanismo de pila dual es esencial para la coexistencia de ambas versiones IP.

Por último se debe de tener en cuenta que las aplicaciones pueden trabajar en el mejor de los casos con ambas versiones, aplicaciones duales, si no sólo con IPv4 o sólo con IPv6. Para los dos últimos escenarios, sería necesario implementar mecanismos de traducción si se requiere tener comunicación en equipos que sólo soporten una versión del protocolo.

7.3. Inventario y situación de servicios y aplicaciones actuales con IPv4.

En el caso particular de la UNAM, dado que RedUNAM tiene como propósito ofrecer servicios a la comunidad universitaria así como la transmisión de datos entre facultades, institutos, centros de difusión, coordinaciones y demás dependencias que conforman a la universidad. Las principales acciones que se llevan a cabo en la red universitaria son la descentralización de los servicios de cómputo, modernización de las bases de datos, servicios vía RedUNAM, sistematización computalizada para oficinas, programas de capacitación para personal administrativo, modernización de equipos para la investigación, entre otras actividades; que hacen que la red universitaria sea una de las redes educativas de telecomunicaciones digitales de mayor tamaño y más avanzadas en América Latina.

Es por eso que en la actualidad se dispone de una gran cantidad de equipo de red para realizar estas tareas, y de personal que da mantenimiento a dichos equipos. Pero los equipos en su mayoría no cuentan con soporte IPv6 dado que cuando fueron adquiridos no

se tomó en cuenta que soportaran IPv6 como requerimiento primordial. Así también muchos de los equipos que tenían el soporte de la nueva versión del protocolo no contaban con la versión del software para direccionar adecuadamente el tráfico, por lo que la transición a IPv6 de forma nativa no ha sido posible ampliamente, por ello los mecanismos de transición han sido adoptados por algunos usuarios finales para realizar conexiones a las redes IPv6. Sólo algunos administradores cuentan con equipos adecuados para tener conectividad IPv6 de forma nativa, pero estos casos son aislados y no se puede afirmar que RedUNAM ya se encuentre en el proceso de transición a IPv6.

Un punto importante que se debe de mencionar en RedUNAM, es el hecho que los servicios que se ofrecen a la comunidad universitaria como Correo electrónico, Web, Bases de Datos no se encuentran adaptados mayoritariamente para la transición, ya que muchos de los servidores no cuentan con las últimas versiones del software que soportan adecuadamente IPv6. Como es el caso de los servidores Web, ya que en su gran mayoría utilizan Apache versión 1.3.X y estas versiones no cuentan con soporte IPv6, a menos que se aplique un parche no siempre recomendable. En la tabla 7.1 se lista software que se utiliza con frecuencia para dar algún servicio en RedUNAM que cuenta con soporte IPv6 y que también son utilizados por usuarios para tareas cotidianas.

Aplicaciones	Versión	Soporte IPv6
Apache servidor web	Serie 2	Si
PHP	5.3.0	Si
Mysql	5.4	Si
Sendmail	8.14.2	Si
Postfix	2.2	Si
Internet Explorer	7 u 8	Si
Opera	10	Si
Mozilla Firefox	3.07 o superior	Si
Putty	0.60	Si
VLC	1.0	Si

Tabla 7.1 Software que soporta IPv6 de uso en RedUNAM.

7.4. Selección del mecanismo de transición a IPv6 más adecuado

La selección del mecanismo de transición que sea el más apropiado para comenzar a utilizar IPv6, es directamente dependiente de la arquitectura de la red, ya que no hay una recomendación la cual se pueda asegurar que será la óptima para empezar la transición a la nueva versión del protocolo, por lo que se puede utilizar uno o varios mecanismos para poder tener acceso a las redes IPv6.

Los pasos para implementar un mecanismo de pila dual en una infraestructura de red común se proponen que sean:

- Crear y poner en operación una red de pruebas con túneles IPv6 para poder ganar experiencia sobre los conceptos y características de IPv6.
- Evaluar la versión de software de un ruteador en el ambiente de prueba, de preferencia idéntico a los de producción, para ver qué tan estable y robusto es su comportamiento en una red con IPv4 e IPv6 trabajando en forma conjunta.
- Si son estables y el funcionamiento es adecuado, se puede empezar a actualizar los ruteadores de producción y habilitarles IPv6.
- Si se dan problemas como afectaciones en los servicios de producción se puede volver a sólo operar con IPv4 o tratar de obtener versiones del sistema operativo más recientes.

Muchos vendedores de ruteadores cuentan con equipos que soportan IPv6 en software desde hace algún tiempo, y con las experiencias que se han obtenido, tienen una aceptable estabilización de sus productos. Esto es de gran ventaja para los proveedores de Internet que en un futuro deseen proporcionar conectividad IPv6, así no se caería en el dilema del "*huevo y la gallina*", ya que al momento que se demande conectividad IPv6 sus redes tendrán la posibilidad de soportarlo.

La ventaja del funcionamiento de la pila dual reside en que la red es la misma para IPv4 e IPv6. No se necesitan nuevos ruteadores para IPv6, por lo que no hay necesidad de mantener una red compleja, aunque también esto es una desventaja ya que la red, al ser la misma, los defectos o bugs de software pueden afectar los servicios en ambas versiones, que muy probablemente no sucederían si la red estuviera separada. Por lo que se debe de tener en mente que al probar IPv6 en las redes de producción se podría tener un impacto, ya que la pila dual no se encuentra soportada en hardware en la mayoría de los ruteadores duales que son los que encapsulan IPv6 por medio de software.

Por otra parte, existe una ventaja al utilizar túneles IPv6 sobre la infraestructura existente en IPv4, la cual ya está en funcionamiento con túneles sobre ella e IPv6 trabaja sin ningún problema. Los túneles configurados manualmente, no siempre toman el camino óptimo entre dos sitios, donde un salto IPv6 puede realizar varios saltos IPv4. Con túneles automáticos, por ejemplo, 6to4, ver capítulo 3, los ruteadores IPv6 envían tráfico en los túneles sobre el camino IPv4 más eficiente entre dos ruteadores 6to4. El problema se presenta cuando hay interacción con otros ruteadores IPv6, para los cuales el ruteo es impredecible o aún no lo soportan del todo.
La dependencia en la infraestructura IPv4 actual puede también ser una debilidad, por ejemplo, los problemas de software como la denegación de servicio en contra de los ruteadores, también afectaría a los servicios IPv6.

Finalmente es importante recalcar que en IPv6, la definición de rutas, descubrimiento de vecinos, fragmentación, reensamble de paquetes son realizados por los nodos finales, no por los ruteadores intermedios lo que trae varias ventajas sobre IPv4.

7.5. Pruebas previas necesarias para ofrecer servicios con IPv6 a usuarios

Un proceso muy importante en la implementación de todo protocolo y tecnología no utilizada previamente, lo constituye la realización de prueba, en este caso con IPv6, para contar con resultados y ejemplos de los primeros ensayos que se realizan para poder constatar el buen funcionamiento de IPv6 en los servicios y aplicaciones actualmente en uso.

Una de las primeras pruebas se puede realizar es con navegadores Web, para lo cual es necesario ingresar a páginas con conectividad IPv6 como por ejemplo: www.ipv6forum.com, www.ipv6.unam.mx, www.kame.net, otra alternativa será la configurar un servidor Web propio y levantar un sitio con el soporte IPv6

Para hacer pruebas en una red local que no cuenta con direcciones globales válidas, conviene configurar el servidor Web, lo que se puede hacer en varios sistemas operativos, en un segmento aislado con direcciones de documentación para IPv6, por ejemplo, utilizar la dirección de documentación para el servidor Web http://[2001:db8:31:1::1] o usar direcciones de un bloque propio, por ejemplo, http://[2001:448:11:1::1] como se ve en la figuras 7.3. En este ejemplo la conexión al servidor se realizó a través de algún mecanismo de transición que se mencionó en capítulos anteriores.

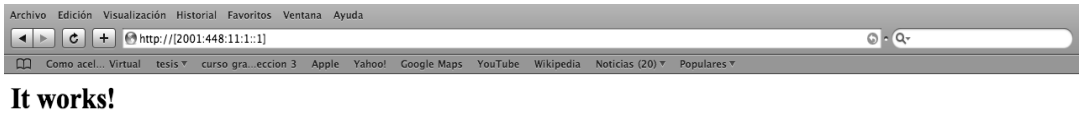


Figura 7.3 Ejemplo de Prueba del soporte IPv6 en navegadores Web.

Así mismo, una prueba sencilla pero que es de gran importancia es saber si los hosts tienen conectividad a las redes IPv6 mediante el envío de paquetes a través de ellas, esto se puede hacer con los paquetes ICMPv6 que se pueden identificar con el Echo Request y Echo Reply,

los cuales son los más comunes de TCP/IP, por medio la herramienta "ping" que existe en las diferentes plataformas. El Ping como se ha mencionado es utilizado para determinar si un host específico está disponible en la red y listo para entablar comunicación. Por lo que básicamente un host emisor envía un mensaje de tipo Echo Request al nodo destino específico y si éste se encuentra disponible envía de vuelta un mensaje del tipo Echo Reply, en la tabla 7.2 se puede ver un ejemplo de un Ping y el envío de paquetes ICMP entre dos equipos usando el mecanismo de túneles con direcciones de documentación desde un equipo con la dirección 2001:db8:31:1::3 a un host con la dirección 2001:db8:31:1::2.

```
#ping6 2001:db8:31:1::2
PING6(56=40+8+8 bytes) 2001:db8:31:1::3 → 2001:db8:31:1::2
16 bytes from 2001:db8:31:1::2, icmp_seq=0 hlim=128 time=0.579 ms
16 bytes from 2001:db8:31:1::2, icmp_seq=4 hlim=128 time=0.563 ms
--- 2001:db8:31:1::2 ping6 statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.563/0.653/0.727/0.068 ms
```

Tabla 7.2 Envío de paquetes ICMPv6 utilizando direcciones de documentación.

Como siguientes pruebas conviene realizar aquellas entre equipos que estén fuera de nuestra red local en segmentos y redes IPv6 separadas y haciendo uso de los distintos mecanismos de transición como los mencionados anteriormente y en el capítulo 3.

Uno de ellos es el denominado 6to4 que pertenece a los mecanismos de transición por túneles, el cual realiza una configuración automática y envía paquetes IPv6 sobre redes IPv4, sus direcciones están formadas por el prefijo 2002::/16 y una dirección IPv4 global única como principal condición, así con la configuración de dos ruteadores en la frontera de una red o entre un ruteador y un host se puede obtener conectividad a las redes IPv6 a nivel global.
En las tablas 7.3 y 7.4 se puede ver la configuración de un ruteador relay 6to4 y la tabla de ruteo, respectivamente en Windows Vista. En los próximos capítulos se describe como hacerlo en diferentes sistemas operativos y como se comportan los equipos que desean acceder a redes IPv6 por este mecanismo.

```
C:\Users\Jose>ipconfig
Configuración IP de Windowsx
Adaptador de Ethernet Conexión de área local:
  Sufijo DNS específico para la conexión. . :
  Dirección IPv6 . . . . . : 2002:84f8:6cec:8:f855:966d:7507:121b
  Vínculo: dirección IPv6 local. . . : fe80::f855:966d:7507:121b%8
  Dirección IPv4. . . . . : 13.24.10.23
  Máscara de subred . . . . . : 255.0.0.0
  Puerta de enlace predeterminada . . . . . : 13.24.10.254
Adaptador de túnel Conexión de área local* 5:
  Sufijo DNS específico para la conexión. . :
  Dirección IPv6 . . . . . : 2002:84f8:6cec::84f8:6cec
  Puerta de enlace predeterminada . . . . . : 2002:c058:6301::c058:6301
```

Tabla 7.3 Configuración de la interfaz de un router relay en Windows Vista.

```
netsh interface ipv6>sh route

Publicar Tipo     Mét  Prefijo                Índ  Puerta enl./Nombre int.
------- -------- ---  ---------------------- ---  ----------------------
No      Manual   256  ::/0                  17  2002:c058:6301::c058:6301
No      Manual   256  ::1/128                 1  Loopback Pseudo-Interface 1
Sí      Manual   1000 2002::/16               17  Conexión de área local*5
No      Manual   0    2002:84f8:6cec:8::/64    8  Conexión de área local
No      Manual   256  2002:84f8:6cec:8::/128   8  Conexión de área local
No      Manual   256  fe80::/64               10  Conexión de área local* 9
No      Manual   256  fe80::100:7f:fffe/128   10  Conexión de área local* 9
No      Manual   256  fe80::b9a1:360d:8636:f8cc/128  15  tun0
No      Manual   256  fe80::f855:966d:7507:121b/128   8  Conexión de área local
No      Manual   0    fec0:0:0:8::/64          8  Conexión de área local
No      Manual   256  fec0:0:0:8::/128         8  Conexión de área local
No      Manual   256  ff00::/8                 1  Loopback Pseudo-Interface1
No      Manual   256  ff00::/8                10  Conexión de área local* 9
```

Tabla 7.4 Rutas del ruteador-relay 6to4 en Windows Vista.

Por su parte Teredo pertenece al mecanismo de transición por túneles que, sirve para brindar conectividad IPv6 a nodos que se encuentran detrás de dispositivos que no reconocen IPv6 como los NATs. Esto lo realiza mediante la encapsulación de paquetes IPv6 dentro de IPv4, permitiendo a los usuarios contar con conectividad IPv6 mientras que su proveedor de servicios de Internet sólo les brinde conectividad IPv4.

Finalmente se puede implementar un Túnel Broker que es un mecanismo por el cual se puede obtener conectividad IPv6 a través de túneles que se configuran de manera automática o manual entre redes o equipos. Como parte de este trabajo de tesis se desarrolló e implementó un servidor de túnel broker, el cual está al servicio de los usuarios de RedUNAM, y que se describe ampliamente en los siguientes capítulos.

7.6. Procedimiento y políticas a seguir para solicitar y utilizar direcciones IPv6

Adicionalmente al implementar y haber probado una nueva tecnología, como IPv6, es importante establecer un procedimiento y una serie de políticas para que los usuarios puedan solicitar y utilizar las direcciones IPv6, mediante los distintos mecanismos de transición probados.

En el caso de la RedUNAM que se encuentra formada por un conjunto de redes locales, tienen una administración independiente, pero que al tener todas salida al Internet a través de la dorsal de RedUNAM, sus encargados y usuarios deben acatar disposiciones que establece la dirección de Telecomunicaciones con representación por parte del NIC-UNAM, lo que representa un esquema jerárquico.

El NIC es el encarga de mantener una Base de Datos donde se concentra la información concerniente a la administración de las redes e instituciones y ofrece las funciones de:

- Servicio de Nombres.
- Asignación de direcciones IP.
- Asignación y solicitud de dominio.
- Servicio de Servidor Secundario.
- Elaboración e Implementación de Políticas.

Los pasos a seguir y las consideraciones para la asignación de direcciones IPv6 en RedUNAM son los siguientes:

1. La solicitud de direcciones IPv6 debe hacerse por medio de su sitio Web o en su defecto, a través de correo electrónico, con el siguiente procedimiento:
Solicitud a través de Web:

- Se ingresa a la página http://www.nic.unam.mx
- Se ingresa al enlace: Formas de registro.
- Se ingresa al enlace: Solicitud de Direcciones IPv6
- Se llena los campos correspondientes a la forma de registro con los datos necesarios.

2.-Este trámite sólo lo podrán realizar los responsables de la red local de la dependencia.

3.- La información contenida en la solicitud es verificada y ratificada de acuerdo a la Base de Datos de NICUNAM. En el caso de encontrar información falsa o no congruente, la solicitud será rechazada y se notificará a los responsables de la dependencia y al solicitante.

4.-En la forma de solicitud se deberá incluir la justificación correspondiente indicando los proyectos en los cuales se emplearán las direcciones IPv6 solicitadas. En caso de no incluirse esta justificación, la solicitud será rechazada inmediatamente.

5.-Las solicitudes serán procesadas de acuerdo al orden en que se reciban.

Si la solicitud cumple con todos los requerimientos se procederá a la asignación de las direcciones IPv6. Una vez asignado el rango de direcciones, los responsables de la red serán los encargados de hacer la distribución de las mismas de acuerdo a su planeación y se encargarán de mantener su relación actualizada, así como los planos correspondientes.

Por lo anterior, NICUNAM establece que:
"Dentro de los servicios que ofrece RedUNAM a la comunidad universitaria, y en particular a las dependencias e instituciones que se conectan a ella, se encuentran: la asignación de direcciones IP; la asignación de dominios; el servicio de nombres; la asignación de dominios inversos; el servicio de servidor secundario, así como el tratamiento de incidentes y quejas de seguridad.

Estos servicios son atendidos por NICunam y para soportarlos existen ciertas políticas que se tienen que seguir con el objetivo de brindar un servicio de calidad. Cada una de estas políticas cuenta con una introducción que ayuda a comprender la naturaleza del servicio de mejor manera. Algunas de ellas se encuentran disponibles y otras, en elaboración.

Estas políticas no son estáticas y se adaptan a las necesidades actuales de recursos de Internet para RedUNAM. Es recomendable que los administradores de redes locales conectadas a RedUNAM conozcan estas políticas y las hagan, a su vez, del conocimiento de sus usuarios. "

Capítulo 8

Pruebas e implantación de Servicios y Aplicaciones con soporte IPv6

8.1. Introducción

Un aspecto importante a considerar para la transición a IPv6, es saber cómo responderán las aplicaciones con las que se trabaja día con día en las redes actuales, ya que para poder hablar de una transición satisfactoria a la nueva versión del protocolo, se tiene que tomar en cuenta que nuestros servicios se puedan ofrecer también con soporte IPv6. Actualmente muchas de las aplicaciones y servicios de los más utilizados como servidores Web, clientes Web, servidores de correo, DNS,SSH, etc. ya cuentan con soporte IPv6, por lo que instalarlos en nuestras redes resulta fácil y su configuración es mínima o nula.

Para probar que tan bueno es una transición en las redes, es necesario realizar una serie de pruebas en las que se puedan ver los contratiempos, ventajas y desventajas, ya que así se podrá saber si se cuenta con las condiciones para adoptar y coexistir con la nueva versión del protocolo y así seleccionar el método más adecuado para llevarlo acabo.

Teniendo en cuenta los tipos de configuraciones con las cuales se puede tener acceso a IPv6 desde un host, las pruebas realizadas se centraron en saber si al conectar un host a un determinado segmento de RedUNAM, éste de forma automática o manual puede adquirir una dirección, y poder idear el mecanismo de transición para darse este proceso de forma transparente para el usuario y de una manera fácil para los administradores de red.

En la actualidad la universidad no cuenta con conexiones automáticas IPv6 para toda la comunidad académica y universitaria, sólo ciertos segmentos cuentan con esta posibilidad, por lo que la conexión en RedUNAM, es casi en su totalidad en IPv4, como se esquematiza en la figura 8.1 y la tabla 8.1.

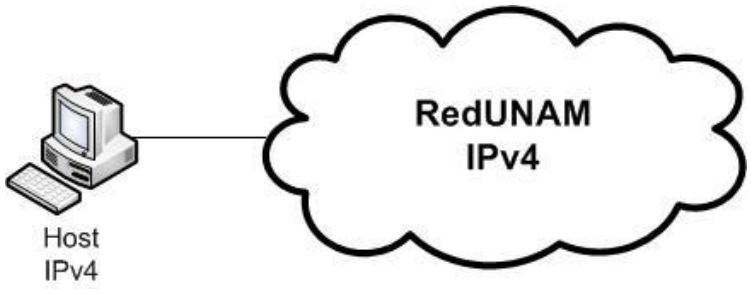


Figura 8.1 Esquema de conexión en RedUNAM

```
Microsoft Windows [Versión 6.0.6001]
Copyright (c) 2006 Microsoft Corporation.  Reservados todos los derechos.

C:\Users>ipconfig
Configuración IP de Windows
Adaptador LAN inalámbrico Conexión de red inalámbrica:
   Estado de los medios. . . . . . . . . . . : medios desconectados
   Sufijo DNS específico para la conexión. . : riu.unam.mx
Adaptador de Ethernet Conexión de área local:
   Sufijo DNS específico para la conexión. . :
   Dirección IPv4. . . . . . . . . . . . . . : 10.0.0.2
   Máscara de subred . . . . . . . . . . . . : 255.0.0.0
   Puerta de enlace predeterminada . . . . . : 10.0.0.1
```

Tabla 8.1 Ejemplo de configuración de un Host en RedUNAM

Muchas de las conexiones en RedUNAM se realizan de forma manual y en algunos casos se implementan mecanismos como NAT y DHCP para realizar las conexiones de forma más rápida, sencilla e intentar prolongar la vida de IPv4.

Es por eso que las pruebas que se han realizado en RedUNAM tienen como objetivo hacer que los usuarios puedan obtener conectividad IPv6 por diferentes mecanismos de transición y con esto realizar una transición más sencilla y eficaz. El esquema general de las pruebas se muestra en la figura 8.2 para aplicar los conocimientos sobre conexiones, métodos de transición y encontrar la mejor opción para los usuarios de RedUNAM.

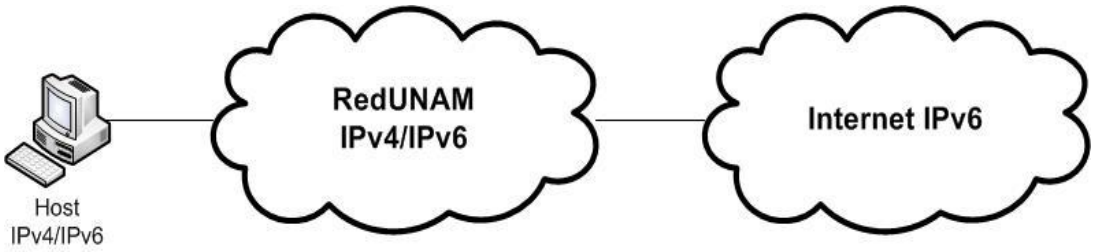


Figura 8.2 Esquema general de pruebas de conectividad con IPv6.

8.2. Configuración de servicios IPv6 en diferentes plataformas.

Como se ha mencionado en capítulos anteriores, la transición efectiva a IPv6 requiere que los servicios cuenten con soporte a la nueva versión del protocolo, y para ello es recomendable observar el comportamiento de dichas aplicaciones con este soporte en un entorno donde existan otros servicios en producción, para así comparar el comportamiento de los que se encuentren funcionando con sólo IPv4 con los que además soporten IPv6, y con ello observar los errores que se puedan generar al implementarlos, buscar cómo corregirlos y llegar a la convivencia de ambas versiones del protocolo en un mismo segmento.

Servicio Web

Uno de los servicios más utilizado es el de Web, ya que en cualquier segmento de red, con la autorización correspondiente, se puede instalar un servidor Web, es por ello que se decidió probar el servidor Web de apache, el cual cuenta con soporte IPv6, puede

implementarse en varias plataformas, y de esta manera poder comprobar en un segmento de red la accesibilidad a un sitio IPv6.

En la tabla 8.2 se listan los sistemas operativos en los cuales se instaló y configuró el servidor web apache.

Sistema operativo	Servicio
Windows XP SP2	Apache web server 2.2.9 con parche
Windows Vista SP1	Apache web server 2.2.9 con parche
Ubuntu 8.04 Hardy Heron	Apache web server 2.2.9
Freebsd 7.0	Apache web server 2.2.13

Tabla 8.2 Sistemas operativos donde se implementó apache

Para la instalación del **servidor web en Windows XP**, se descargó de la página http://www.apachelounge.com una versión de la serie 2 de apache para Windows, en la cual se aplicó un parche para obtener conexión IPv6, ya que los sockets para conexiones IPv6 no funcionan correctamente para los sistemas anteriores a Windows Vista.

En primer lugar se instaló el software del servidor apache, siguiendo los pasos del instalador como se puede ver en la figura 8.3:

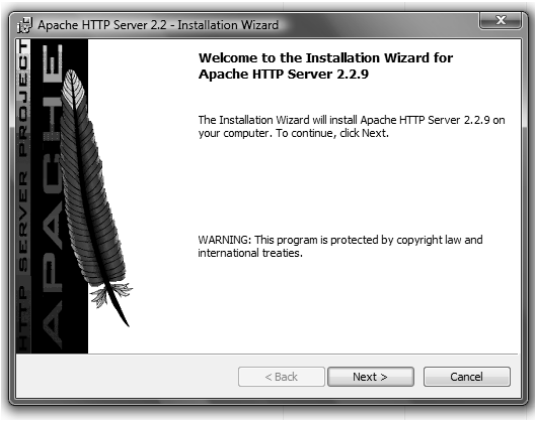


Figura 8.3 Instalador del servidor apache en Windows.

Al concluir con la instalación es muy importante realizar la configuración correspondiente para probar el servicio. Para lograr que el servidor apache pudiera realizar conexiones a través de direcciones IPv6, se configuró el archivo conf/httpd.conf, como se ve en la tabla 8.3, agregando la sentencia *Listen [::]* y el puerto por el cual el servidor apache escuchará las peticiones.

```
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost> directive.
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen [::]
```

Tabla 8.3 Fragmento del archivo de configuración httpd.conf en Windows XP.

Con esta configuración se puede iniciar el servicio de apache a través de una línea de comandos de Windows la sentencia C:\..\Apache\httpd.exe –k start

Con esta simple configuración se habilita el servidor web en Windows XP en un segmento de red.

Para comprobar su funcionamiento se conectó al mismo segmento de red un cliente con soporte IPv6 habilitado, y utilizando diferentes navegadores Web se estableció una conexión al servidor como se ve en las figuras 8.4 y 8.5 donde se utilizó el navegador Web Firefox y safari respectivamente para realizar la prueba.

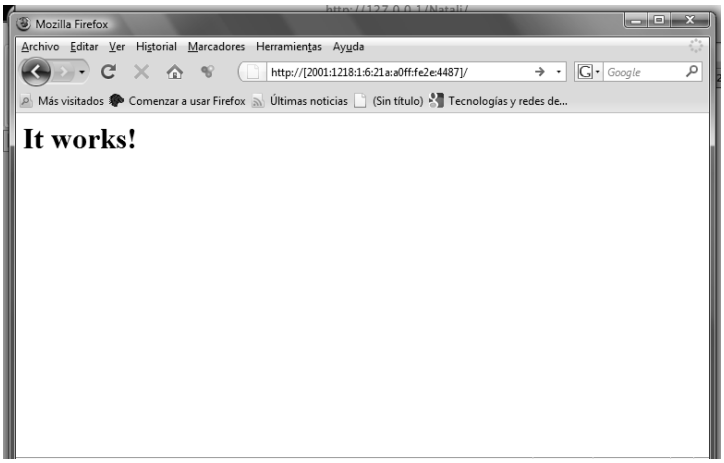


Figura 8.4 Prueba de conexión al servidor Web en Windows XP a través del navegador firefox.

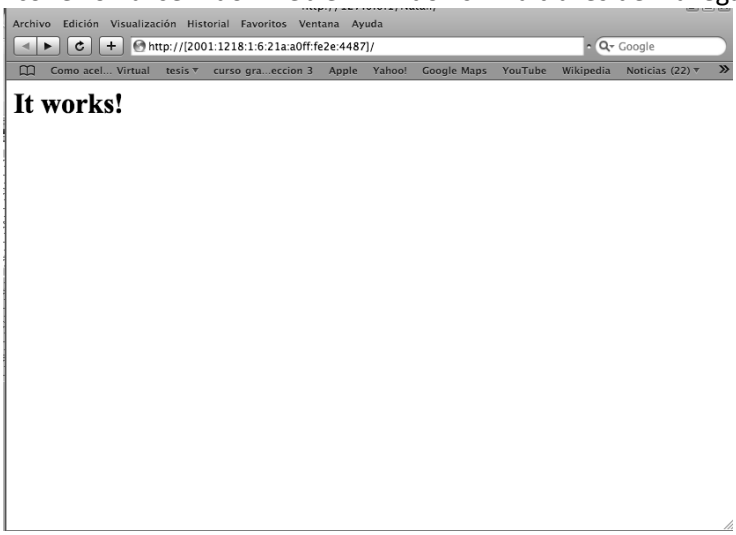


Figura 8.5 Prueba de conexión al servidor web en Windows XP a través del navegador safari.

Para la instalación del **servidor web en Windows Vista**, se utilizó la misma versión de apache con soporte IPv6 para Windows de http://www.apachelounge.com, y se ejecutó el mismo instalador. Al terminar la instalación el siguiente paso fue la configuración del servicio para comprobar su funcionamiento con IPv6. Al igual que en Windows XP la configuración se realiza sobre el archivo conf/httpd.conf pero a diferencia de Windows XP en Windows Vista sólo se agrega el número de puerto de escucha como se muestra en la tabla 8.4, y sólo con eso el servidor Web puede ofrecer conexiones en IPv4 e IPv6.

```
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80
```

Tabla 8.4 Fragmento del archivo de configuración httpd.conf en Windows Vista.

Como se muestra en las figuras 8.6 y 8.7, también se realizó la prueba de conexión al servidor web instalado en Windows Vista a través de clientes web.

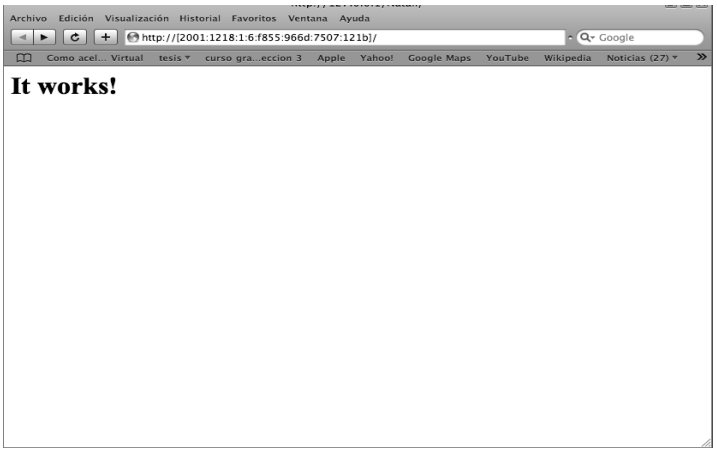


Figura 8.6 Prueba de conexión al servidor Web en Windows Vista a través del navegador Safari.

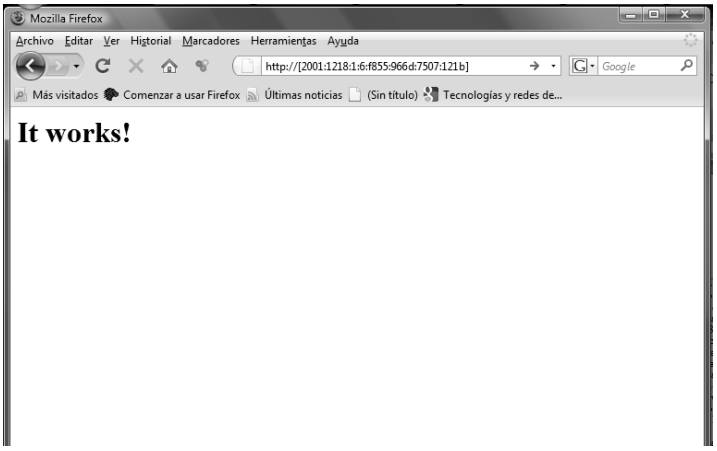


Figura 8.7 Prueba de conexión al servidor Web en Windows Vista a través del navegador Firefox.

La implementación del servidor web, también se realizó en una distribución **Ubuntu 8.04 Hardy Heron**. El servidor apache se descargó de la página oficial de apache http://httpd.apache.org/download.cgi, por lo que se continuó con la instalación del servicio con la ejecución de los siguientes comandos:

```
$ sudo gzip -d httpd-2.2.9.tar.gz
$ sudo tar xvf httpd-2.2.9.tar        Se descomprime el servidor apache.
$ cd httpd-2.2.9                      Se introduce en la carpeta creada
$ sudo ./configure                    Se ejecuta el script configure.
$ sudo make                           Se  compila el servidor web.
$ sudo make install                   Se instala el servidor web.
```

Después del proceso de instalación se procedió a modificar el archivo de configuración de apache /usr/local/apache2/conf/httpd.conf para agregar la sentencia Listen [::] y para asegurar el puerto de escucha a través de IPv6, como se muestra en la tabla 8.5.

```
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen [::]
```

Tabla 8.5 Fragmento del archivo de configuración httpd.conf en servidor Ubuntu.

Al igual que en los sistemas Windows se realizó la prueba utilizando un host conectado en el mismo segmento de red y accediendo al servidor web con la ayuda de navegadores web como se muestra en las figuras 8.8 y 8.9.

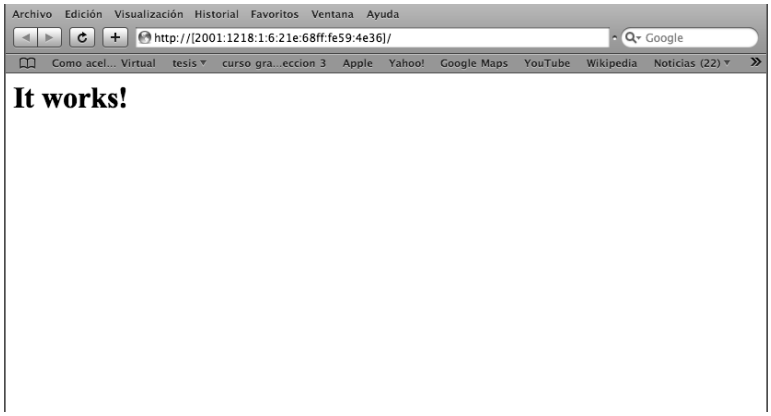


Figura 8.8 Prueba de conexión al servidor web en Linux a través del navegador Safari.

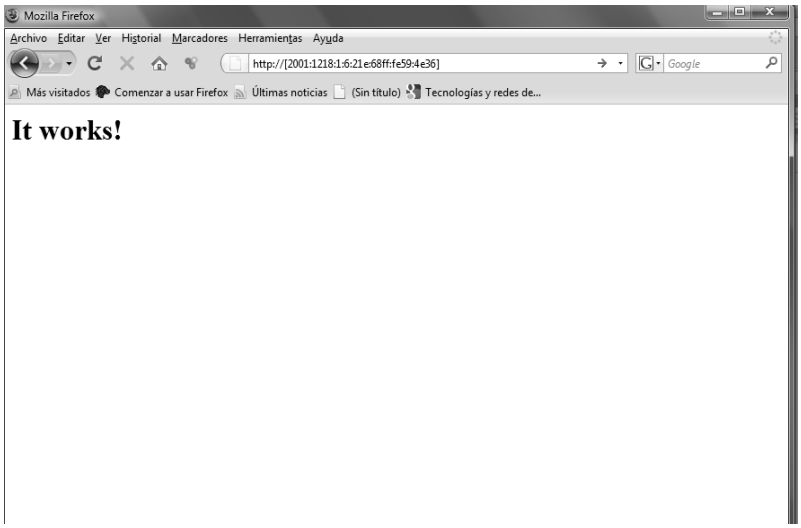


Figura 8.9 Prueba de conexión al servidor web en Linux a través del navegador Firefox.

Finalmente la instalación de un servidor Web en un **sistema FreeBSD** se realizó con la descarga de la última versión del servidor web apache de la página http://httpd.apache.org/download.cgi, por lo que se continuó con la instalación ejecutando los siguientes comandos:

$ sudo tar xvf httpd-2.2.13.tar	Se descomprime el servidor apache.
$ cd httpd-2.2.13	Se introduce en la carpeta creada
$ ./configure	Se ejecuta el script configure.
$ make	Se compila el servidor web.
$ make install	Se instala el servidor web.

Esta versión del servidor apache ya escucha por defecto en ambas versiones del protocolo de Internet por lo que no es necesario modificar el archivo /usr/local/apache2/conf/httpd.conf sólo se modificaría el puerto por el cual se desea acceder al servidor como se muestra en la tabla 8.6.

```
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80
```

Tabla 8.6 Fragmento del archivo de configuración httpd.conf en servidor FreeBSD.

Por lo que para comprobar que el servidor web estuviera en funcionamiento desde hosts conectados en el mismo segmento se accedió utilizando navegadores web, como se muestra en las figuras 8.10 y 8.11.

Figura 8.10 Prueba de conexión al servidor web en FreeBSD a través del navegador Safari.

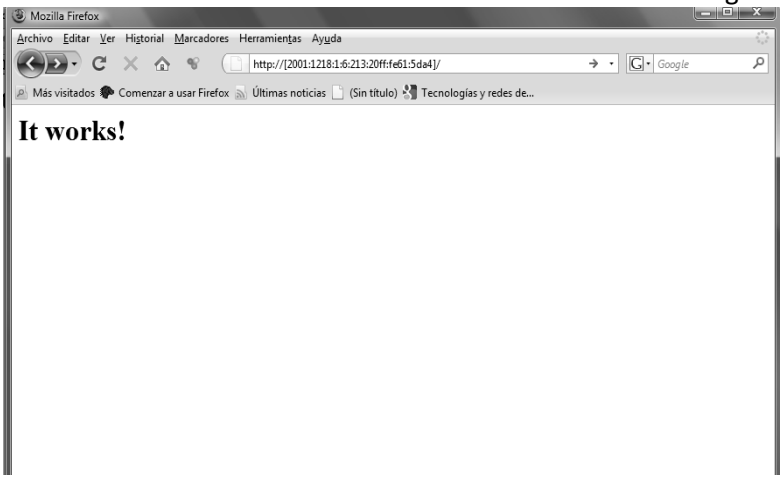


Figura 8.11 Prueba de conexión al servidor web en FreeBSD a través del navegador Firefox.

Relay 6to4

Es un servicio que es posible habilitar en las redes que desean realizar pruebas de IPv6 sobre IPv4 mediante un relay 6to4, el cual puede brindar conectividad IPv6 a una red o un host, y así poder aplicar conceptos, prueba de servicios y facilitar la transición a la nueva versión del protocolo.

Un relay 6to4 se puede configurar en un sistema operativo FreeBSD, para realizar pruebas de conectividad de un host a través de este mecanismo de transición.

En **FreeBSD** se debe tomar en cuenta que estos sistemas hacen uso de la interfaz *stf* para este propósito, por lo que la versión del kernel debe ser capaz de soportar dicha interfaz para la implementación de este método. Para implementar el relay 6to4 se debe habilitar en el archivo rc.conf el reenvío de paquetes agregando la sentencia ***ipv6_enable="YES"*** lo aplica a todas las interfases que se utilicen en el equipo lo pueden realizar. Los comandos para la configuración de 6to4 se muestra en la tabla 8.7 donde se incluye la dirección que se asigna a la interfaz stf así como la ruta para los paquetes con el prefijo del mecanismo de transición.

```
# ifconfig stf create
# ifconfig stf inet6 2002:84f8:6cee:1::1/16
# route add -inet6 2002::/16 -interface stf
```

Tabla 8.7 Configuración de la interfaz 6to4 en FreeBSD.

En los sistemas **Windows XP y Vista**, el servicio de 6to4 se encuentra habilitado por defecto, por lo que estos sistemas ya cuentan con una interfaz especial para este mecanismo de transición.
Para poder brindar conectividad a través de un relay 6to4 en los sistemas operativos Windows se ejecutan los comandos que se muestran en la tabla 8.8.

```
C:\netsh interface ipv6 6to4 set state state=enabled
Aceptar
C:\netsh interface ipv6 6to4 set routing routing=enabled
Aceptar
```

Tabla 8.8 Configuración de un relay 6to4 en Windows XP y Vista.

6to4 también se puede implementar en ruteadores, por mencionar un ejemplo de la marca **Cisco** que soporten este mecanismo de transición, esto se realiza con una configuración sencilla como se muestra en la tabla 8.9 y así puede tener habilitado el servicio de túneles.

```
!
interface Tunnel2002
ipv6 address 2002:c058:7387::1/16
tunnel source 13.24.10.25
tunnel mode ipv6ip 6to4
!
Ipv6 route ::/0
```

Tabla 8.9 Ejemplo de configuración de 6to4 en un ruteador Cisco.

Esta configuración dotará al ruterador con direcciones 6to4 en una de sus interfases y dado que esta conectado a una red IPv6, podrá dar conectividad a los host que lo utilicen como relay.

Teredo

Como se mencionó en el capítulo anterior, Teredo es un mecanismo de transición que brinda conectividad IPv6 a dispositivos que se encuentren detrás de un NAT, por lo que es necesario de un servidor Teredo asigne una dirección IPv6 a dicho dispositivo y con la ayuda de un Teredo Reley le de acceso a otros equipos o sitios IPv6.
Para este servicio se utilizo el software Miredo de la página http://www.remlab.net/miredo/ el cual se instaló en un equipo con sistema operativo Debian. Después de la descarga del software se ejecutaron los siguientes comandos para la instalación del servidor Teredo:

# tar xvf miredo-X.Y.tar.bz2	Se descomprime el software, donde X.Y. es la ultima versión de miredo.
# cd miredo-X.Y.Z	Se accede a la carpeta creada.
# ./configure	Se ejecuta el script configure.
# make	Se compila el software.
# make install	Se instala el servidor teredo.

Uno de los requerimientos para montar un servidor teredo, es la necesidad de tener dos direcciones IPv4 consecutivas en el mismo equipo para el buen funcionamiento del servidor, así que se configura el archivo /etc/network/interface y se agrega una nueva interfaz como un alias como se muestra en la tabla 8.10.

```
/etc/network$ more interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
#allow-hotplug eth0 eth0:1
auto eth0 eth0:1
iface eth0 inet static
   address 13.24.10.24
   netmask 255.0.0.0
   network 13.24.10.224
   broadcast 13.24.10.255
   gateway 13.24.10.254
   # dns-* options are implemented by the resolvconf package, if installed
   dns-nameservers
   dns-search ipv6.unam.mx
iface eth0:1 inet static
   address 13.24.10.25
  netmask 255.0.0.0
   network 13.24.10.224
   broadcast 13.24.10.255
```

Tabla 8.10 Configuración de la interfaz alias para servidor teredo.

Para configurar el servidor teredo se debe modificar el archivo etc/miredo/miredo-server.conf agregando la dirección del equipo que será el servidor teredo como se muestra en la tabla 8.11.

```
netlab@netlab:/etc/miredo$ more miredo-server.conf
# Please refer to the miredo-server.conf(5) manpage for details.

# Server primary IPv4 address.
# Miredo will open UDP port 3544 on this IPv4 address and the next one.
ServerBindAddress 13.24.10.24
```

Tabla 8.11 Archivo de configuración del servidor teredo.

Al finalizar la instalación y configuración el servidor teredo este puede asignar direcciones a los clientes que se encuentren detrás de dispositivos como NAT que deseen conectarse a través de IPv6 por este mecanismo de transición, en la tabla 8.12 se muestra la configuración de las interfases del servidor.

```
netlab@netlab:~$ /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:13:20:74:12:3c
          inet addr: 13.24.10.24  Bcast: 13.24.10.255 Mask:255.0.0.0
          inet6 addr: 2001:1218:1:6:213:20ff:fe74:123c/64 Scope:Global
          inet6 addr: fe80::213:20ff:fe74:123c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:74 errors:0 dropped:0 overruns:0 frame:0
          TX packets:61 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4014 (3.9 KiB)  TX bytes:8557 (8.3 KiB)
          Interrupt:17

eth0:1    Link encap:Ethernet  HWaddr 00:13:20:74:12:3c
          inet addr: 13.24.10.25  Bcast: 13.24.10.255  Mask: 255.0.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:17

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: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)

teredo    Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet6 addr: fe80::ffff:ffff:ffff/64 Scope:Link
          UP POINTOPOINT RUNNING NOARP  MTU:1280  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 B)  TX bytes:144 (144.0 B)
```

Tabla 8.12 Interfases del servidor teredo.

Así mismo los clientes que deseen conectarse a través del servidor teredo en particular deben realizar cambios a la configuración de red como es el caso de Windows Vista que se lleva a cabo mediante el comando netsh en una ventana de comandos ejecutando las siguientes instrucciones como se muestra en la tabla 8.13.

```
C:\>netsh
Netsh> interface ipv6
Netsh interface ipv6>set teredo client 13.24.10.24
```

Tabla 8.13 Configuración de un cliente teredo en Windows Vista.

Por lo que un cliente después de configurar el mecanismo de transición teredo puede obtener una dirección como se muestra en la tabla 8.14 y acceder a las redes IPv6.

```
C:\Users\Jose>ipconfig
Configuración IP de Windows
Adaptador de Ethernet Conexión de área local 3:
   Estado de los medios. . . . . . . . . . . : medios desconectados
   Sufijo DNS específico para la conexión. . :
Adaptador de Ethernet Conexión de área local:
   Sufijo DNS específico para la conexión. . : undesirable-ipv6.netlab.unam.mx
   Vínculo: dirección IPv6 local. . . : fe80::f855:966d:7507:121b%8
   Dirección IPv4. . . . . . . . . . . . . . : 192.168.108.51
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : 192.168.108.254
Adaptador de túnel Conexión de área local*:
   Estado de los medios. . . . . . . . . . . : medios desconectados
   Sufijo DNS específico para la conexión. . : undesirable-ipv6.netlab.unam.mx
Adaptador de túnel Conexión de área local* 9:
   Sufijo DNS específico para la conexión. . :
   Dirección IPv6 . . . . . . . . . . . : 2001:0:53aa:64c:24f4:78f8:7b07:9315
   Vínculo: dirección IPv6 local. . . : fe80::24f4:78f8:7b07:9315%10
   Puerta de enlace predeterminada . . . . . : ::
```

Tabla 8.14 Interfases de Windows Vista después de configurar el mecanismo de transición teredo.

Para un cliente teredo que se encuentra detrás de un NAT con un sistema operativo Linux, que utiliza una distribución de Ubuntu Hardy Heron, es necesario descargar el software de la página http://www.remlab.net/miredo/ e instalarlo como los siguientes comandos:

# tar xvf miredo-X.Y.tar.bz2	Se descomprime el software, donde X.Y. es la ultima versión de miredo.
# cd miredo-X.Y.Z	Se accede a la carpeta creada.
# ./configure	Se ejecuta el script configure.
# make	Se compila el software.
# make install	Se instala miredo.

Se continua configurando el software para que pueda realizar la conexión al servidor teredo, obtener una dirección unicast global y pueda realizar conexiones a las redes IPv6, la configuración se realiza modificando el archivo /etc/miredo.conf como se muestra en la tabla 8.15.

```
/etc$ vi miredo.conf
RelayType restricted
# Please refer to the miredo.conf(5) man page for details.
InterfaceName    teredo
# Pick a Teredo server:
#ServerAddress    teredo.ipv6.microsoft.com
#ServerAddress    teredo-debian.remlab.net
ServerAddress 13.24.108.24

# Some firewall/NAT setups require a specific UDP port number:
#BindPort    3545
```

Tabla 8.15 Configuración de un cliente teredo en Linux.

Para que el host pueda utilizar el mecanismo de transición teredo, en el archivo de configuración se debe agregar la dirección IP del servidor que provee la dirección unicast global que se necesita para la conexión a través de IPv6 así que se agrega la línea ServerAddress "**IPv4 del Servidor**" y declarar el tipo de NAT en el cual se encuentra conectado el host en la línea RelayType restricted, hecho esto las interfases en un sistema operativo Ubuntu se muestran en la tabla 8.16.

```
ath0      Link encap:Ethernet  dirección HW 00:1f:e1:37:35:c2
          inet dirección:10.4.250.242  Difusión:10.4.250.255  MÃ¡scara:255.255.255.0
          dirección inet6: fe80::21f:e1ff:fe37:35c2/64 Alcance:VÃnculo
          ARRIBA DIFUSIÃ“N CORRIENDO MULTICAST  MTU:1500  MÃ©trica:1
          RX packets:1870 errors:0 dropped:0 overruns:0 frame:0
          TX packets:970 errors:2 dropped:2 overruns:0 carrier:0
          colisiones:0 txqueuelen:0
          RX bytes:1494154 (1.4 MB)  TX bytes:129525 (126.4 KB)
lo        Link encap:Bucle local
          inet dirección:127.0.0.1  MÃ¡scara:255.0.0.0
          dirección inet6: ::1/128 Alcance:Anfitrión
          ARRIBA LOOPBACK CORRIENDO  MTU:16436  MÃ©trica:1
          RX packets:1647 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1647 errors:0 dropped:0 overruns:0 carrier:0
          colisiones:0 txqueuelen:0
          RX bytes:86368 (84.3 KB)  TX bytes:86368 (84.3 KB)
teredo    Link encap:UNSPEC  direcciónHW 00-00-00-00-00-00-00-00-00-00-00-00-00-00-
00-00
          dirección inet6: fe80::ffff:ffff:ffff/64 Alcance:Vinculo
          dirección inet6: 2001:0:84f8:6cf0:8c4:6caa:7b08:a1e/32 Alcance:Global
          ARRIBA PUNTO A PUNTO CORRIENDO NOARP  MTU:1280  MÃ©trica:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
          colisiones:0 txqueuelen:500
          RX bytes:416 (416.0 B)  TX bytes:816 (816.0 B)
```

Tabla 8.16 Interfases en Ubuntu después de configurar el mecanismo de transición teredo.

Servidor de túneles broker

Otro servicio que se implementó es el de un **servidor de túneles broker**, el cual ayuda a usuarios de RedUNAM a configurar una conexión IPv6 mediante un túnel IPv6 sobre IPv4 realizando una solicitud vía Web. Con la ejecución de un script que configura automáticamente la conexión a través del túnel o de forma manual por medio de comandos.

Para poner en marcha este servicio, el servidor debe contar con software que soporte IPv6 el cual se lista en la tabla 8.17 y se implemento en un equipo con las características que se muestran en la tabla 8.18.

Software	Versión
PHP	5.3.0
Servidor web apache	2.2.13
PHPMyAdmin	2.11.22
MySQL	5.0

Tabla 8.17 Software necesario para implementar el servido de túneles broker.

Sistema Operativo	Procesador	Memoria RAM
FreeeBSD 7.0	Intel Pentium 4 3.4GHz	1GB

Tabla 8.18 Características del equipo utilizado para el servido de túneles broker.

Para la instalación de PHP se descargó la última versión del software de la página http://www.php.net/downloads.php y se ejecutaron los siguientes comandos:

$ tar xvf php-5.3.0.tar.gz	Se descomprime el software
$ cd php-5.3.0	Se accede a la carpeta del software.
$ ./configure	Se ejecuta el script configure.
$ make	Se compila el software.
$ make install	Se instala PHP.

Anteriormente se describió cómo instalar el servidor web apache, por lo que se sigue el mismo procedimiento de instalación para implementar el servidor de túneles bróker, por lo que se ejecutándose los siguientes comandos:

$ sudo tar xvf httpd-2.2.13.tar	Se descomprime el servidor apache.
$ cd httpd-2.2.13	Se introduce en la carpeta del servidor
$ ./configure	Se ejecuta el script configure.
$ make	Se compila el servidor web.
$ make install	Se instala el servidor web.

La instalación de PHPMyAdmin se puede realizar de una forma fácil en los sistemas operativos BSD a través de los ports del sistema los cuales se encuentran ubicados en /usr/ports y donde se encuentran muchos de los programas precompilados y listos para su instalación.

Así para la instalación se ejecutaron los comandos que se muestran a continuación:

#cd /usr/ports/databases/phpmyadmin	Se accede a la carpeta de los ports donde se encuentra phpmyadmin.
#make install clean	Se instala phpmyadmin.

El último paquete que es necesario para la implementación del túnel broker es mysql que es de ayuda para gestionar la base de datos utilizada para obtener los datos de cada usuario y poder configurar de forma correcta los túneles, así como llevar un control de los mismos y brindar conectividad IPv6.

De la misma forma se accedió a los ports de FreeBSD y se ejecutaron los siguientes comandos:

#cd /usr/ports/databases/mysql50-server	Se accede a la carpeta del port donde se encuentra MYSQL
#make install clean	Se instala MYSQL.

Después de instalar el software necesario se realizan modificaciones a los archivos de configuración de algunos de ellos para que los programas puedan interactuar de forma correcta. El primer software a configurar es el servidor apache que modificando el archivo /usr/local/apache2/conf/httpd.conf se consigue la interacción con el resto de los programas.

En primer lugar se debe de definir el puerto por el cual el servidor apache escuchará las peticiones, y como se mencionó con anterioridad, éste se hace agregando la línea:

Listen 8080

En seguida se agrega el módulo de PHP que utilizará Apache, para lo cual se agrega la siguiente línea:

LoadModule php5_module modules/libphp5.so

También se deben de configurar las extensiones de las páginas que puede reconocer PHP con la siguiente sentencia:

DirectoryIndex index.php index.html index.htm

Así mismo Apache debe saber con qué debe analizar los archivos PHP y esto se define con las siguientes líneas:

AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps

Y para habilitar PHP en el sistema se puede copiar y renombrar el archivo usr/local/etc/php.ini-dist en /usr/local/etc/ y con esto tener la configuración de PHP, para esto se ejecuta el siguiente comando:

$ cp /usr/local/etc/php.ini-dist /usr/local/etc/php.ini

Instalado y configurado el software listado con en la figura 8.1 se procedió a programar un sistema en el cual se almacenen los datos de los interesados en obtener conectividad IPv6 a través de este servicio y la configuración de cada uno de los túneles hechos por los usuarios. El objetivo del servidor se lleva a cabo gracias a la interacción de la programación HTML, PHP, Shell y a la obtención de datos mediante consultas SQL a la base de datos, para así llevar acabo la configuración de los túneles y tener en funcionamiento este mecanismo de transición.

Como resultado final se obtuvo una interfaz como se muestra en la figura 8.12 la cual puede ser consultada desde cualquier parte de RedUNAM para el beneficio de sus usuarios que deseen realizar conexiones con IPv6.

Figura 8.12 Página principal del Servidor de túnel Broker implementado.

CONCLUSIONES

A partir del análisis y los datos recabados sobre la situación de RedUNAM dado que no se cuenta con soporte IPv6 en su Backbone, lo que ha frenado la transición a IPv6 de forma nativa a la red universitaria, se constató que los mecanismos de transición son esenciales para ir implementando la nueva versión del protocolo en algunos nodos y redes en operación de la universidad hoy en día. Por tal motivo se llevaron a cabo pruebas en diversos sistemas operativos, tomando en cuenta que muchos de éstos son de los más utilizados por los usuarios y laboratorios de la DGSCA. Algunos problemas presentados se debieron a que el software de las aplicaciones que se encuentran en algunos equipos no es el óptimo para poner en marcha algún servicio público con soporte IPv6 ya que es necesario actualizarlo, así como encontrar las dependencias o parches adecuados para que el funcionamiento de la implementación sea el idóneo. Así mismo, se probó el funcionamiento de algunos métodos de transición para ofrecer conectividad IPv6 a la comunidad universitaria para ayudar a la coexistencia y transición requerida.

Los servicios que finalmente se implementaron para ofrecer conectividad IPv6 en RedUNAM, de la forma más transparente para los usuarios son mediante un servidor teredo, un Relay 6to4 y un servidor de túneles broker.

Por una parte dado que el mecanismo de Túnel teredo es utilizado cuando los hosts se encuentran detrás de dispositivos NAT, como es el caso de la RIU, se implementó un servidor Teredo que les asigna direcciones IPv6 públicas, ayudando a levantar un túnel IPv6 en IPv4 y mediante los equipos Relays, externos a RedUNAM, se les da conectividad IPv6 sin dejar de tenerla con IPv4.

Por otra parte los mecanismos de túnel 6to4 y túnel Broker se implementaron en diversos equipos y sistemas operativos del laboratorio y se obtuvieron resultados satisfactorios, por su buen comportamiento para ofrecer conectividad IPv6 a usuarios de RedUNAM. Las pruebas realizadas y los resultados son mostradas en los Anexos 1 y 2, donde se describen las alternativas que los administradores y usuarios interesados en tener un acercamiento a la nueva versión del protocolo puedan utilizar en sus redes que en su mayoría son sólo con soporte IPv4.

Al haber concluido el trabajo de investigación y las pruebas propuestas, las contribuciones de esta tesis fueron:

- Material de referencia teórico sobre conceptos IPv6 y de cómo se puede obtener conectividad mediante los mecanismos implementados en RedUNAM, con ello adquirir conocimientos sólidos sobre la nueva versión del protocolo.
- Un servidor de túneles Broker que puede ser usado por académicos, administradores y la comunidad en general dentro de RedUNAM para obtener conectividad IPv6 por medio de scripts de configuración automática.
- Habilitación del mecanismo 6to4 en el ruteador del laboratorio para ofrecer conectividad IPv6 mediante éste.
- El desarrollo de scripts para ofrecer:
    - Configuración de túneles IPv6 punto a punto entre equipos.
    - Configuración de forma automática por algún mecanismo de transición implementado para esta investigación.
    - Estadísticas y datos de la configuración IPv6 actual en un equipo.

Los mecanismos de transición de túnel teredo, túnel 6to4 y túnel broker, son implementaciones que se encuentran funcionando actualmente en el NETLab (Laboratorio de Tecnologías Emergentes de Red); los administradores y usuarios de RedUNAM que deseen tener un primer acercamiento a la nueva versión del protocolo de Internet, pueden obtener conectividad mediante éstos, para que en un futuro los costos e impactos que deje la puesta en servicio y utilización de IPv6 en sus propias redes sea menor al contar con experiencia previa.

A partir de esta tesis se pueden realizar diversos trabajos sobre IPv6, ya que sienta las bases para la investigación de los temas que no son abordados en la misma, teniendo en común la utilización de la nueva versión del protocolo de Internet. Ya que los objetivos de este trabajo fueron el dar a conocer el estado de algunos de los servicios con soporte IPv6 que se utilizan en Internet, dar una guía para la configuración de IPv6 que servirá a los administradores para que puedan ofrecer conectividad, comiencen la coexistencia y la transición a la nueva versión del protocolo, con ello puedan prepararse para el agotamiento de las direcciones IPv4 disponibles.

Algunas de las actividades pendientes recomendadas incluyen:

- Actualización del túnel broker, ya que los mecanismos de transición pueden sufrir modificaciones con el paso del tiempo y así seguir brindando conectividad IPv6.
- Generación de scripts para la configuración automática de IPv6 en diferentes sistemas operativos utilizando diversos lenguajes de programación.
- Diseño e implementación de uno o varios servidores de Nombre de Dominio que puedan funcionar con IPv6 de forma nativa en RedUNAM.
- Implementación de software de ruteo que soporte IPv6, para poder distribuir prefijos mediante diversos protocolos de encaminamiento.
- Implementación y varias pruebas de equipos de seguridad para ambas versiones del IP.

Referencias.

- Hagen Silvia. (2002). "*IPV6 essentials".* Beijing. O'Reilly.
- Gallo, Michael A., Hancock William W. (2002). "*Comunicación entre computadoras y tecnologías de redes*". Madrid. Thomson.
- Niall Richard Murphy, David Malone. (2005). "*IPv6 Network Administration*". United states of America, O'Reilly.
- Iljitsch van Beijnum (2006), "*Running IPv6*", California, Apress.
- John Amoss, Daniel Minoli. (2008). "*Handbook of IPv4 to IPv6 transition: Methodologies for Institutional and Corporate Networks*". United States of America. Auerbach Publications.
- Mark A. Sportack. (2003). "*Fundamentos de enrutamiento IP*". Madrid. Pearson Educacion.
- Craig Zacker. (2002) "*Redes: manual de referencia*". Madrid-México. McGraw-Hill Interamericana.
- Mark A. Sportack. (2003). **"***Fundamentos de enrutamiento IP***".** Madrid. Pearson Educacion.
- Regis Desmeules. (2003). **"***Cisco self-study: implementing IPv6 networks (IPV6)***".** Indianapolis, Indiana. Cisco.
- Pete Loshin. (2004). **"***IPv6: theory, protocol, and practice***"**. San Francisco, California. M. Kaufmann.

Mesografía.

- Página IPv6 de la UNAM, <http://www.ipv6.unam.mx/>
- IPv6 - Transición a IPv6 de América Latina y el Caribe | Portal de Transición a IPv6 de América Latina y el Caribe. <http://portalipv6.lacnic.net/>
- Wiki del Grupo de Trabajo IPv6, Red clara. <http://wiki-gtipv6.reuna.cl/wiki/index.php/Portada>
- Capítulo Mexicano del Foro IPv6. <http://www.ipv6forum.com.mx/>
- IPv6, Forum Driving IPv6 Deployment. <http://www.ipv6forum.com>
- The IPv6 portal. <http://www.ist-ipv6.org/index.php>
- IPv6 Day. <http://www.ipv6day.org/action.php?n=Es.IPv6day>

- Dirección de Telecomunicaciones, RedUNAM. <http://www.dgsca.unam.mx/dtd/redunam.html>
- Centro de Información de RedUNAM. <http://www.nic.unam.mx/>
- Centro de Operación de la Red. <http://www.noc.unam.mx/>
- Documentación del servidor de HTTP Apache. <http://httpd.apache.org/docs/2.2/>
- Miredo: Teredo IPv6 tunneling for Linux and BSD. <http://www.remlab.net/miredo/>
- Documentación: Manual de Freebsd. <http://www.freebsd.org/es/docs.html>

RFC

[1] K. Ramakrishnan, S. Floyd, D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP" RFC 3168, Septiembre2001. <www.ietf.org/rfc/rfc3168.txt>

[2] S. Bradner, A. Mankin, "The Recommendation for the IP Next Generation Protocol", RFC 1726, Enero 1995. <http://www.ietf.org/rfc/rfc1752.txt>

[3] C. Partridge, F. Kastenholz, "Technical Criteria for Choosing IP The Next Generation (IPng)", RFC 1726, Diciembre 1994. <http://www.ietf.org/rfc/rfc1726.txt>

[4] Jon Postel, "Internet Protocol", RFC 791, Septiembre 1981, <http://www.ietf.org/rfc/rfc0791.txt>

[5] J. Mogul, Jon Postel, "Internet Standard Subnetting Procedure", RFC 950, Agosto1985, <http://www.faqs.org/rfcs/rfc950.html>

[6] Jeffrey Mogul, "Broadcasting Internet Datagrams in the Presence of Subnets", RFC 922, Octubre 1984. <http://www.rfc-editor.org/rfc/rfc922.txt>

[7] K. Nichols, S. Blake, F. Baker, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, Diciembre 1998. <http://www.ietf.org/rfc/rfc2474.txt>

[8] R. Hinden, S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, Febrero 2006. <http://www.ietf.org/rfc/rfc4291.txt>

[9] C. Partridge, T. Mendez, W. Milliken, "Host Anycasting Service", RFC 1546, Noviembre 1993, <http://www.ietf.org/rfc/rfc1546.txt>

[10] R. Hinden, S. Deering, "IPv6 Multicast Address Assignments", RFC 2375, Julio 1998. <http://www.ietf.org/rfc/rfc2375.txt>

[11]R. Hinden, S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, Julio 1998. <http://www.ietf.org/rfc/rfc2373.txt>

[12] R. Gilligan, E. Nordmark, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, Octubre 2005. <http://www.ietf.org/rfc/rfc4213.txt>

[13] Conta, S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, Diciembre 1998. <http://www.ietf.org/rfc/rfc2473.txt>.

[14] Carpenter, C. Jung, "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels", RFC 2529, Marzo 1999. <http://www.ietf.org/rfc/rfc2529.txt>

[15] Carpenter, K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, Febrero 2001. <http://www.ietf.org/rfc/rfc3056.txt>

[16] K. Egevang, P. Francis, "The IP Network Address Translator (NAT)", RFC 1631, Mayo 1994, <http://www.ietf.org/rfc/rfc1631.txt>

[17] J. Rosenberg, J. Weinberger, C.Huitema, R. Mahy, "Session Traversal Utilities for NAT", RFC 5394, <http://www.ietf.org/rfc/rfc5394.txt>

[18] Huitema, "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, Febrero 2006. <http://www.ietf.org/rfc/rfc4380.txt>

[19]Durand, P. Fasano, I. Guardini, D. Lento, "IPv6 Tunnel Broker", RFC 3053, Enero 2001. http://www.ietf.org/rfc/rfc3053.txt

[20] Templin, T. Gleeson, M. Talwar, D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 4214, Octubre 2005. <http://www.ietf.org/rfc/rfc4214.txt>

[21]R. Coltun, D. Ferguson, J. Moy, "OSPF for IPv6", RFC 2740, Julio 2008. <http://www.ietf.org/rfc/rfc5340.txt>

[22] Y. Rekhter, "A Border Gateway Protocol 4 (BGP-4)" , RFC 4271, Enero 2006. <http://www.ietf.org/rfc/rfc4271.txt>

[23]T. Bates, R. Chandra, D. Katz, Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, Enero 2007. <http://tools.ietf.org/html/rfc4760>

[24]S. Thomson, C. Huitema, V. Ksinant, M. Souissi, "DNS Extensions to Support IP Version 6", RFC 3596, Octubre 2003. <http://www.ietf.org/rfc/rfc3596.txt>

[25] M. Blanchet, "A Flexible Method for Managing the Assignment of Bits of an IPv6 Address Block", RFC 3531, Abril 2003. <http://www.ietf.org/rfc/rfc3531.txt>

[26] R. Hinden, S. Deering, "IP Version 6 Addressing Architecture", RFC 1884, Diciembre 1995. <http://www.ietf.org/rfc/rfc1884.txt>

[27]P. Mockapetris, "DOMAIN NAMES - CONCEPTS AND FACILITIES", RFC 1034, Noviembre 1987. <http://www.ietf.org/rfc/rfc1034.txt>

[28] S. Thomson, C. Huitema, "DNS Extensions to support IP version 6", RFC 1886, Diciembre 1995. <http://www.ietf.org/rfc/rfc1886>

[29] M. Crawford, C. Huitema, "DNS Extensions to Support IPv6 Address Aggregation and Renumbering", RFC 2874, Julio 2000. <http://www.rfc-editor.org/rfc/rfc2874.txt>

[30]M. Crawford, "Binary Labels in the Domain Name System", RFC 2673, Agosto 1999. <http://www.ietf.org/rfc/rfc2673.txt>

[31]M. Crawford," Non-Terminal DNS Name Redirection", RFC 2672, agosto 1999. <http://www.ietf.org/rfc/rfc2672.txt>

[32] R. Bush, A. Durand, B. Fink, T.Hain, "Representing Internet Protocol version 6 (IPv6) Address in the Domain Name System", RFC 3363, Agosto 2002. <http://www.faqs.org/rfcs/rfc3363.html>

[33] R.Austein, "Tradeoffs in Domain Name System (DNS) Support for Internet Protocol version 6 (IPv6)", RFC 3364, Agosto 2002. < http://ftp.eenet.ee/doc/rfc/rfc3364.txt>

[34] G. Huston, "Management Guidelines & Operational Requirements for the Address and Routing Parameter Area Domain ("arpa")", RFC 3172, Septiembre 2001. <http://www.rfc-editor.org/rfc/rfc3172.txt>

[35]N. Walsh, J. Cowan, P. Grosso, "A URN Namespace for Public Identifiers", RFC 3152, Agosto 2001. <http://www.ietf.org/rfc/rfc3151.txt>

[36] R. Draves, "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3483, Febrero 2003. <http://www.rfc-editor.org/rfc/rfc3484.txt>

[37]D. Eastlak, "Domain Name System Security Extensions", RFC 2535, Marzo 1999. <http://www.ietf.org/rfc/rfc2535.txt>

[38]P. Vixie, O. Gudmundsson, D. Eastlake, B. Wellington, "Secret Key Transaction Authentication for DNS" RFC 2845, Mayo 2000. <http://www.ietf.org/rfc/rfc2845.txt>

ANEXO 1

I. Habilitación o instalación y configuración manual de IPv6 en diferentes plataformas.

En este anexo, se menciona cómo instalar o simplemente habilitar y configurar IPv6 en las diferentes plataformas utilizadas y probadas en el desarrollo de este trabajo que fueron:

- Windows XP SP2 o superior.
- Windows Vista SP1.
- Ubuntu Hardy Heron 8.04
- Debian 5.0
- FreeBSD 7.0.

Así se pretende dar una guía con la cual se pueda tener una visión clara sobre los procedimientos a seguir en los equipos con diferentes sistemas operativos para instalar, habilitar y configurar IPv6, ofrecer alternativas para la transición a la nueva versión del protocolo de Internet y tener acceso a redes IPv6 mediante servicios configurados en la universidad.

A) Implementación de IPv6 en plataforma Windows XP

Desde que la IETF definió la nueva versión del protocolo de Internet, que se dió a conocer con el nombre de IPv6 el MSR por sus siglas en inglés (Microsoft Research) ha contribuido a la estandarización desde 1996 a 2002 para sus sistemas operativos, con lo cual se creó el proyecto MSR-IPv6, el cual tenía como objetivo brindar simplicidad, seguridad y movilidad IPv6. A principios de 1998 liberaron la primer versión de la aplicación que añadía nuevas características para soporte IPv6 para sistemas como Windows NT/2000.

Habilitación de IPv6 en Windows XP

En los sistemas Operativos Windows XP una forma de habilitar el soporte IPv6, es abriendo el símbolo del sistema como administrador, seleccionando el botón de **Inicio > ejecutar** y escribir el comando **cmd;** una vez abierto se ejecuta el siguiente comando:

- Netsh interface ipv6 install

Así se habilitará en el sistema el soporte de IPv6.

Verificación del soporte IPv6

Para verificar rápidamente los cambios en las interfases que se dieron con la habilitación de IPv6 se ejecuta el siguiente comando en la ventana de símbolo del sistema:

- Ipconfig /all

Ya que se habilitó el soporte IPv6 y se verificó que en las interfases se ha configurado una dirección de enlace local, estas direcciones sólo tienen sentido entre las interfases de nuestra propia red, por lo que para poder tener conectividad a Internet IPv6 se debe de tener una dirección IPv6 global, con la cual se podrá tener acceso a diferentes servicios fuera de la red local.

Otra forma con la cual se puede comprobar que el soporte IPv6 está instalado y habilitado es que en una ventana de comandos se ejecute el comando ping ::1, para hacer un ping a la dirección de loopback IPv6 y se obtendrá un resultado como se muestra en la tabla A.1.

```
C:\>ping ::1
Haciendo ping a ::1 desde ::1 con 32 bytes de datos:
Respuesta desde ::1: tiempo<1m
Respuesta desde ::1: tiempo<1m
Respuesta desde ::1: tiempo<1m
Estadísticas de ping para ::1:
Paquetes: enviados = 3, recibidos = 3, perdidos = 0
(0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Tabla A.1 Ping a la dirección de Loopback después de habilitar/instalar el soporte IPv6.

Configuración manual de interfases con IPv6 en Windows XP

Para obtener una dirección IPv6 global, se puede hacer de dos formas, manual y automática. La forma automática, como se mencionó en el capitulo 2 es cuando un ruteador o DHCP asigna una dirección IPv6.

Para el caso de la asignación manual de una dirección IPv6 a una interfaz, se utiliza el comando netsh el cual está precargado en los sistemas Windows, con este comando se puede realizar la configuración de las interfases de la red. La configuración de una dirección a una interfaz se muestra en la tabla A.2.

```
C:\netsh
netsh> interface
netsh interface> ipv6
netsh interface ipv6> add address [interface=] <nombre interface> [address=]<dirección
IPv6>[[type=]unicast | anycast]
Ejemplo:
C:\netsh
netsh> interface
netsh interface> ipv6
netsh interface ipv6> add address interface="Local Area Conection"address=2001:db8:31::1 type=unicast
```

Tabla A.2 Configuración manual de una dirección IPv6 en un interfaz de Windows.

Otro dato importante que se puede y debe configurar de forma manual es la ruta que seguirán los paquetes IPv6, la definición de una ruta por default o una ruta por una determinada interfaz, esto se realiza ejecutando los comandos que se muestran en la tabla A.3.

```
C:\netsh
netsh> interface
netsh interface> ipv6
netsh interface ipv6> add route [prefix=]<dirección IPv6>/<entero> [interface=]<cadena>
    [[nexthop=]<dirección IPv6>]
Ejemplo:
C:\netsh
netsh> interface
netsh interface> ipv6
netsh interface ipv6> add route prefix=::/0 interface="Local Area Conection" nexthop=2001:db8:31::2
```

Tabla A.3 Configuración manual de rutas IPv6 en Windows.

B) Implementación de IPv6 en plataforma Windows Vista.

Los equipos con el sistema operativo Vista cuentan con el soporte IPv6 instalado y habilitado por defecto, con esto se puede experimentar con uno o varios métodos de transición sin la necesidad de realizar una configuración anterior.
Microsoft Windows Vista incluye un buen soporte de IPv6, no sólo de características básicas como ocurre en anteriores versiones de Windows como Windows XP y 2003, sino también características avanzadas como:

- Doble pila IPv4/IPv6 instalada y habilitada por defecto.
- Configuración basada en interfaz gráfica de usuario.
- Soporte completo para IPSec .
- MLDv2.
- LLMNR.
- Direcciones IPv6 literales en las URLs.
- Soporte de IPv6 en conexiones PPP.
- DHCPv6 .
- Identificadores de interfaz aleatorios.

Verificación del soporte IPv6

Para verificar que el sistema cuente con el soporte IPv6 habilitado se ejecuta el comando ping ::1 en una ventana de símbolo del sistema como se muestra en la tabla A.4.

```
Microsoft Windows [Versión 6.0.6001]
Copyright (c) 2006 Microsoft Corporation.  Reservados todos los derechos.
C:\>ping ::1
Haciendo ping a ::1 desde ::1 con 32 bytes de datos:
Respuesta desde ::1: tiempo<1m
Respuesta desde ::1: tiempo<1m
Respuesta desde ::1: tiempo<1m
Respuesta desde ::1: 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
```

Tabla A.4 Verificación de soporte IPv6 en Windows Vista.

Configuración manual de interfases con IPv6 en Windows Vista

Como se había mencionado con el comando netsh se pueden configurar las interfases en los sistemas Windows, para agregar direcciones, rutas y métodos de transición, se realizan con sólo ejecutar las instrucciones correspondientes en los diferentes contextos de este comando, como se mostró en las tablas A.2 y A3.

C) Implementación de IPv6 en plataforma Ubuntu Hardy Heron 8.04 y Debian

La primera implementación IPv6 en Linux fue hecha en Noviembre de 1996 para el *kernel* 2.1.8 basada en el API de BSD, las distribuciones Ubuntu cuentan con el soporte IPv6 instalado y habilitado por defecto, con lo cual se puede implementar uno o varios mecanismos de transición.

Verificación del soporte IPv6

Para verificar que el sistema cuente con soporte IPv6 se ejecuta un ping a la dirección de loopback de IPv6 y para saber si el sistema soporta o no la nueva versión del protocolo, el resultado debe ser como el que se muestra en la tabla A.5.

```
$ ping6 ::1
PING6(56=40+8+8 bytes) ::1 --> ::1
64 bytes from ::1, icmp_seq=0 hlim=64 time=0.381 ms
64 bytes from ::1, icmp_seq=1 hlim=64 time=0.443 ms
64 bytes from ::1, icmp_seq=2 hlim=64 time=0.495 ms
--- ::1 ping6 statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.381/0.440/0.495/0.047 ms
```

Tabla A.5 Verificación del soporte IPv6 en Ubuntu Hardy Heron y Debian.

Configuración manual de interfases con IPv6 en Ubuntu y Debian

La configuración de una interfaz en las distribuciones Linux como se ha mencionado con anterioridad, se puede realizar de forma automática o manual, por lo que a través de una *terminal* y con ayuda del comando "ip" se puede agregar la información necesaria a una interfaz, como se muestra en la tabla A.6.

```
$ ip address{ add | del } <dirección IPv6> />prefijo> dev <nombre de la interfaz>
Ejemplo:
$ ip address add 2001:db8:11:1::1/64 dev eth0
```

Tabla A.6 Configuración de una dirección IPv6 en Ubuntu y Debian.

La configuración de rutas por defecto por la cual se pueden envíar los paquetes IPv6, con el comando "route" o especificar por que interfaz se desea hacer el envío de los paquetes, se realiza con la instrucción que se muestra en la tabla A.7.

```
$ route -A inet6 { add | del }  <prefijo |default> gw <dirección del siguiente salto>
Ejemplo:
$ route -A inet6 add default gw 2001:448:20:29::56
```

Tabla A.7 Configuración de rutas IPv6 en Ubuntu y Debian.

D) Implementación de IPv6 en plataforma FreeBSD 7.01

En sus versiones más recientes el kernel de FreeBSD cuenta con soporte IPv6 instalado y habilitado por defecto, por lo que no tiene problema con la conectividad IPv6, automáticamente crean dirección de enlace local en todas las interfases, pero no así su dirección global por autoconfiguración, a través del mecanismo *"Sin estado",* para lo cual se debe de habilitar esta opción modificando el archivo /etc/rc.conf, agregando las siguientes líneas y reiniciando el equipo.

ipv6_enable="YES"
ipv6_network_interfaces="auto"

Verificación del soporte IPv6

Para verificar que el sistema FreeBSD soporta IPv6 se ejecuta un ping a la dirección de loopback IPv6 y así saber si el sistema soporta o no la nueva versión del protocolo, el resultado debe ser como el que se muestra en la tabla A.8.

```
# ping6 ::1
PING6(56=40+8+8 bytes) ::1 --> ::1
16 bytes from ::1, icmp_seq=0 hlim=64 time=0.381 ms
16 bytes from ::1, icmp_seq=1 hlim=64 time=0.390 ms
16 bytes from ::1, icmp_seq=2 hlim=64 time=0.490 ms
--- ::1 ping6 statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.381/0.390/0.490 ms
```

Tabla A.8 Verificación del soporte IPv6 en FreeBSD.

Configuración manual de Interfases con IPv6 en FreeBSD

Para configurar una dirección IPv6 de forma manual en una interfaz de FreeBSD, puede realizarse a través de una terminal del sistema con ayuda del comando "ifconfig" y los privilegios necesarios para la ejecución de los comandos, que se muestra en la tabla A.9.

```
# ifconfig <nombre de la interfaz> inet6 <dirección IPv6> <prefijo | eui64>
Ejemplo:
# ifconfig gif0 inet6 2001:db8:31:1:: eui64
```

Tabla A.9 Configuración manual de una dirección IPv6 en una interfaz de FreeBSD.

Habiendo configurado una dirección IPv6 en el sistema es posible configurar una ruta por la cual viajarán los paquetes a través de una interfaz, esto se puede realizar manualmente con el comando "route", el cual, a partir de una serie de parámetros como se muestra en la tabla A.10 puede realizar esta configuración.

```
# route add inet6 <interfaz | dirección IPv6> <Siguiente salto | dirección IPv6>
Ejemplo:
# route add inet6 default 2001:db8:31:1::1
```

Tabla A.10 Configuración de rutas IPv6 en FreeBSD.

II. Configuración de túneles.

A) En forma manual.

Windows XP/ Windows Vista

Como se ha mencionado la instalación de IPv6 en los sistemas Windows puede realizarse a través de la línea de comandos, así también la creación de túneles sólo se puede realizar usando el comando *netsh* a través de una línea de comandos como se muestra en la tabla A.11.

```
C:\>netsh
netsh>interface ipv6
netsh interface ipv6>add v6v4tunnel interface=tunel0 localaddress=192.0.2.1
remoteaddress=192.0.2.2
netsh interface ipv6>add address interface=tunel0 address=2001:db8:31:1::1
Aceptar.
netsh interface ipv6>add route prefix=::/0 interface=tunel0 nexthop=2001:db8:31:1::2
Aceptar.
netsh interface ipv6>bye
```

Tabla A.11 Ejemplo de instrucciones para la creación y configuración manual de un túnel en sistemas Windows.

Cuando se ejecuta *netsh* sin argumentos, éste queda a la espera de ejecutar comandos, en este caso estos comandos son los contextos *interface ipv6.* Después de cambiar a este contexto el objetivo es crear el túnel, para esto se tiene la instrucción *add v6v4tunnel.* La opción *interface* es con la que se define el nombre del túnel, en este ejemplo *tunel0.* Las opciones *localaddress* y *remoteaddress* especifican la dirección local y remota IPv4 que serán usadas por el túnel y son llamados puntos finales del túnel. El siguiente comando agrega una dirección a la nueva interfaz. Finalmente en la tercera instrucción es necesario crear una ruta por defecto, con la opción *nexthop* utilizando una dirección IPv6 que puede ser un ruteador remoto o el equipo con el cual se quiere establecer comunicación mediante el túnel; por lo que al final de la configuración el host tiene una nueva interfaz con la dirección IPv6 configurada como se observa en la tabla A.12.

```
C:\ >ipconfig
Configuración IP de Windows
Adaptador LAN inalámbrico Conexión de red inalámbrica:
   Sufijo DNS específico para la conexión. . :
   Vínculo: dirección IPv6 local. . . : fe80::d087:c07a:716e:50cf%9
   Dirección IPv4. . . . . . . . . . . . . . : 192.0.2.1
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : 192.0.2.2

Adaptador de túnel tunel0:
   Sufijo DNS específico para la conexión. . :
   Dirección IPv6 . . . . . . . . . . . : 2001:db8:31:1::1
   Puerta de enlace predeterminada . . . . . : 2001:db8:31:1::2
```

Tabla A.12 Interfaz de túnel configurada en un sistema Windows.

Verificación de la conectividad IPv6

Cuando se configura entre dos equipos esta clase de túneles se puede verificar la conectividad entre ambos extremos con la ayuda del comando ping, y observar cual es el camino de los paquetes ICMP con el comando tracert como se muestra en las tablas A.13 y A14.

```
C:\>ping 2001:db8:31:1::2
Haciendo ping a 2001:db8:31:1::2 desde 2001:db8:31:1::1 con 32 bytes de datos:
Respuesta desde 2001:db8:31:1::2: tiempo=1ms
Tiempo de espera agotado para esta solicitud.
Respuesta desde 2001:db8:31:1::2: tiempo=1ms
Respuesta desde 2001:db8:31:1::2: tiempo<1m
Estadísticas de ping para 2001:db8:31:1::2:
   Paquetes: enviados = 4, recibidos = 3, perdidos = 1
   (25% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 0ms, Máximo = 1ms, Media = 0ms
```

Tabla A.13 Ejemplo de prueba de conectividad entre hosts por medio del túnel.

```
C:\>tracert 2001:db8:31:1::2
Traza a 2001:db8:31:1::2 sobre caminos de 30 saltos como máximo.
  1    3 ms    2 ms    6 ms  2001:db8:31:1::2
Traza completa.
```

Tabla A.14 Ejemplo de traza de los paquetes a través del túnel configurado.

Linux

Aunque hay varias maneras de establecer un túnel manual en Linux, la forma más adecuada para obtener los mejores resultados es cuando se utiliza el comando "ip". En la tabla A.15 se presentan los comandos para crear y configurar un túnel con el comando *ip.*

```
# ip tunnel add name tun0 mode sit local 192.0.2.1 remote 192.0.2.2 ttl 64
# ip link set dev tun0 up
# ip address add 2001:db8:31:1::2/64 dev tun0
```

Tabla A.15 Ejemplo de creación y configuración de un túnel en Linux.

El modo de *sit* se refiere a la transición simple de Internet (Simple Internet Transition). El tipo de túnel *sit* es usado para todos los túneles IPv6 en IPv4, incluyendo 6to4. La diferencia entre los túneles 6to4 y los túneles configurados es la dirección remota. Para que sea posible utilizar el túnel es necesario agregar una ruta por defecto para los paquetes que viajarán por el túnel. Esto se puede realizar como se muestra en la tabla A.16.

```
# route -A inet6 add default gw 2001:db8:31:1::1
```

Tabla A.16 Ejemplo de configuración de una ruta por defecto para el túnel manual en Linux.

Verificación de la conectividad IPv6

Después de haber configurado el túnel entre dos equipos se pude verificar la conectividad entre equipos con el comando ping6 y la ruta que siguen los paquetes por el túnel, como se muestran en las tablas A.17 y A.18.

```
# ping6 2001:db8:31:1::1
PING 2001:db8:31:1::1 (2001:db8:31:1::1) 56 data bytes
64 bytes from 2001:db8:31:1::1: icmp_seq=1 ttl=64 time=0.481 ms
64 bytes from 2001:db8:31:1::1: icmp_seq=2 ttl=64 time=0.455 ms
64 bytes from 2001:db8:31:1::1: icmp_seq=3 ttl=64 time=0.465 ms

--- 2001:db8:31:1::1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.455/0.467/0.481/0.010 ms
```

Tabla A.17 Ejemplo de prueba de conectividad entre hosts por medio del túnel en Linux.

```
# traceroute6 2001:db8:31:1::1
traceroute to 2001:db8:31:1::1 (2001:db8:31:1::1), 30 hops max, 40 byte packets
 1  2001:448:12:1::2 (2001:db8:31:1::1)  4.887 ms  4.864 ms  4.940 ms
```

Tabla A.18 Ejemplo de traza de los paquetes a través del túnel configurado en Linux.

FreeBSD

En FreeBSD, los túneles son implementados bajo la interfaz virtual *gif.* Una interfaz *gif* es configurada con dos direcciones IPv4: el origen del túnel y el destino del túnel. Esto puede hacerse con la ayuda del comando *"ifconfig"* y la interfaz *gif* puede ser configurada como cualquier otra interfaz. En la tabla A.19 se muestra cómo crear la interfaz *gif* y la configuración del túnel.

```
# ifconfig gif3
# ifconfig gif3 tunnel 192.0.2.1 192.0.2.2
gif3: flags=8050<POINTOPOINT,RUNNING,MULTICAST> mtu 1280
tunnel inet 192.0.2.1 --> 192.0.2.2
# ifconfig gif3 up
# ifconfig gif3 inet6 2001:db8:31:1::1
# ifconfig gif3
gif3: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1280
tunnel inet 192.0.2.1 --> 223.224.225.226
inet6 fe80::201:2ff:fe29:2640%gif0 prefixlen 64 scopeid 0x9
inet6 2001:db8:31:1:201:2ff:fe29:2640 prefixlen 64
```

Tabla A.19 Ejemplo de configuración de túnel en FreeBSD para una interfaz *gif.*

Algo importante es que se necesita una interfaz con un estado "*up"* para que los siguientes comandos no arrojen un mensaje de error por la falta de una dirección de enlace local y así se pueda configurar la dirección en la interfaz del túnel. Al configurar la interfaz en el sistema, éste agrega automáticamente entradas a la tabla de ruteo sobre el prefijo de la dirección y hacia la interfaz correspondiente, pero aun así es necesario configurar una ruta por defecto para que los paquetes

viajen a través de la interfaz del túnel, como se muestra en la tabla A.20.

```
# route add -inet6 default 2001:db8:31:1::2
# netstat -rnf inet6
Routing tables
Internet6:
Destination       Gateway           Flags     Netif  Expire
default           2001:db8:31:1::2  UG1c       gif3
```

Tabla A.20 Ejemplo de configuración de una ruta por defecto en FreeBSD.

Verificación de la conectividad IPv6

Después de haber realizado la configuración del túnel se puede verificar la conectividad entre hosts mediante el comando ping6 y verificar la ruta por la cual viajan los paquetes mediante traceroute6, como se muestra en las tablas A.21 y A.22.

```
$ ping6 2001:db8:31:1::2
PING6(56=40+8+8 bytes) 2001:db8:31:1::1--> 2001:db8:31:1::2
16 bytes from 2001:db8:31:1::2, icmp_seq=0 hlim=128 time=0.419 ms
16 bytes from 2001:db8:31:1::2, icmp_seq=1 hlim=128 time=0.667 ms
--- 2001:db8:31:1::2 ping6 statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/std-dev = 0.419/0.543/0.667/0.124 ms
```

Tabla A.21 Prueba de conectividad entre hosts por medio del túnel en FreeBSD.

```
$ traceroute6 2001:db8:31:1::2
traceroute6 to 2001:db8:31:1::2 (2001:db8:31:1::2) from 2001:db8:31:1::1, 64 hops max, 12 byte
packets
 1  2001:db8:31:1::20.335 ms  0.142 ms  0.123 ms
```

Tabla A.22 Traza de los paquetes a través del túnel configurado en FreeBSD.

B) En forma automática mediante 6to4.

Como se vió en el capitulo 8 se puede configurar un relay 6to4 el cual puede dar conectividad IPv6 a clientes que configuren este servicio a través de él, a continuación se describe la forma en que se debe de configurar el servicio en diferentes sistemas operativos y obtener conectividad IPv6 mediante este mecanismo.

Windows XP y Windows Vista

En Windows no es necesario realizar explícitamente la habilitación de 6to4, ya que al realizar la habilitación/instalación de IPv6 en los sistemas Windows XP, si se tiene una dirección IPv4 pública entonces el sistema levanta y configura automáticamente la pseudo interfaz 6to4; en los sistemas operativos Windows Vista, IPv6 se encuentra instalado y habilitado por defecto por lo que también se encuentra la interfaz 6to4 habilitada por defecto.

Para hacer uso del relay 6to4 de la UNAM, en los sistemas operativos Windows se deben ejecutar los comandos que se muestran en la tabla A.23.

```
C:\>netsh
netsh> interface ipv6 6to4
netsh interface ipv6 6to4>set relay 132.248.108.254
Aceptar
```

Tabla A.23 Configuración de 6to4 relay de la UNAM en Windows.

Un ejemplo de las interfases en Windows después de la configuración de 6to4 se muestra en la tabla A.24.

```
C:\>ipconfig
Configuración IP de Windows
Adaptador LAN inalámbrico Conexión de red inalámbrica:
  Sufijo DNS específico para la conexión. . :
  Vínculo: dirección IPv6 local. . . : fe80::d087:c07a:716e:50cf%9
  Dirección IPv4. . . . . . . . . . . . . . : 132.248.214.94
  Máscara de subred . . . . . . . . . . . . : 255.255.255.192
  Puerta de enlace predeterminada . . . . . : 132.248.214.126
Adaptador de túnel Conexión de área local* 5:
  Sufijo DNS específico para la conexión. . :
  Dirección IPv6 . . . . . . . . . . . : 2002:84f8:d65e::84f8:d65e
  Puerta de enlace predeterminada . . . . . : 2002:84f8:6cfe::1
```

Tabla A.24 Ejemplo de configuración de las interfases en Windows.

Verificación de la conectividad IPv6

En un segmento de RedUNAM donde sólo se cuenta con IPv4, habiendo configurado 6to4 se puede verificar la conectividad IPv6 con los comandos ping y tracert como se observa en las tablas A.25 y A.26.

```
C:\>ping -6 www.ipv6forum.com
Haciendo ping a www.ipv6forum.com [2001:a18:1:20::22] desde 2002:84f8:d65e::84f8
:d65e con 32 bytes de datos:
Respuesta desde 2001:a18:1:20::22: tiempo=611ms
Respuesta desde 2001:a18:1:20::22: tiempo=630ms
Respuesta desde 2001:a18:1:20::22: tiempo=629ms
Respuesta desde 2001:a18:1:20::22: tiempo=618ms
Estadísticas de ping para 2001:a18:1:20::22:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0
   (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 611ms, Máximo = 630ms, Media = 622ms
```

Tabla A.25 Prueba de conectividad a través de un relay 6to4 de la UNAM.

```
C:\>tracert -6 www.ipv6forum.com
Traza a la dirección www.ipv6forum.com [2001:a18:1:20::22]
sobre un máximo de 30 saltos:
  1   160 ms    *      158 ms  2001:448::2d0:58ff:fef3:6d41
  2   133 ms   115 ms   119 ms  2001:1228:11b:90a::1
  3   113 ms   116 ms   114 ms  2001:1228:10a:f09::1
  4   113 ms   117 ms   113 ms  2001:1228:10a:f02::2
  5   284 ms    *      321 ms  abilene-1-lo-jmb-702.lsanca.pacificwave.net [2001:504:b:20::131]
  6   318 ms   284 ms   302 ms  so-0-0-0.0.rtr.hous.net.internet2.edu [2001:468:ff:304::1]
  7   114 ms   130 ms   117 ms  xe-1-0-0.0.rtr.atla.net.internet2.edu [2001:468:ff:103::2]
  8   161 ms   164 ms   143 ms  xe-2-0-0.0.rtr.wash.net.internet2.edu [2001:468:ff:109::2]
  9   216 ms   217 ms   215 ms  abilene-wash.rt1.fra.de.geant2.net [2001:798:14:10aa::11]
 10   232 ms   232 ms   231 ms  restena-gw.rt1.fra.de.geant2.net [2001:798:14:10aa::22]
 11   454 ms   231 ms   230 ms  te-5-4.gate-2.bce.restena.lu [2001:a18:0:102::2]
 12   240 ms   236 ms   231 ms  gate-1-v26.rest.restena.lu [2001:a18:0:ff01::1]
 13   233 ms   235 ms   240 ms  fw.restena.lu [2001:a18:0:408::4]
 14   232 ms   237 ms   333 ms  www.ipv6forum.org [2001:a18:1:20::22]
Traza completa.
```

Tabla A.26 Traza de los paquetes utilizando el relay 6to4 de la UNAM.

FreeBSD

FreeBSD usa la interfaz *stf* para los túneles 6to4, a partir de FreeBSD 5.2 ésta se crea con el comando *ifconfig stf create*. Así esta interfaz puede recibir una dirección IPv6 válida para el mecanismo de 6to4 que corresponde a una dirección IPv4 pública de un host. Para habilitar el relay 6to4 de la UNAM en FreeBSD se ejecutan los comandos de la tabla A.27 con los privilegios necesarios.

```
# ifconfig stf0 create
# ifconfig stf0 inet6 2002:84f8:6cee::1/16
# route add -inet6 default 2002: 84f8:6cfe::1
```

Tabla A.27 Configuración de 6to4 en FreeBSD a través de un relay de la UNAM.

Un ejemplo de las interfases en FreeBSD después de la configuración de 6to4 se muestra en la tabla A.28.

```
# ifconfig
msk0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=19a<TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM,TSO4>
        ether 00:13:20:61:5d:a4
        inet6 fe80::213:20ff:fe61:5da4%msk0 prefixlen 64 scopeid 0x1
        inet 13.24.10.23 netmask 0xfffffffe0 broadcast 13.24.21.255
        media: Ethernet autoselect (10baseT/UTP <half-duplex>)
        status: active
stf0: flags=1<UP> metric 0 mtu 1280
        inet6 2002:84f8:6cee::1 prefixlen 16
```

Tabla A.28 Interfases de FreeBSD después de la configuración de 6to4.

Verificación de la conectividad IPv6

Después de configurar la interfaz del túnel 6to4 al relay de la UNAM se pueden realizar pruebas de conectividad con los comandos ping6 y traceroute6 a un sitio IPv6 como se muestra en las tablas A.29 y A.30.

```
$ ping6 www.ipv6forum.com
PING6(56=40+8+8 bytes) 2002:84f8:6cee:1::1 --> 2001:a18:1:20::22
16 bytes from 2001:a18:1:20::22, icmp_seq=0 hlim=128 time=1.212 ms
16 bytes from 2001:a18:1:20::22, icmp_seq=1 hlim=128 time=0.808 ms
16 bytes from 2001:a18:1:20::22, icmp_seq=2 hlim=128 time=0.736 ms
```

Tabla A.29 Prueba de conectividad a través de un relay 6to4 de la UNAM en FreeBSD.

```
# traceroute6 to www.ipv6forum.com (2001:a18:1:20::22) from 2002:84f8:6cee:1::1, 64 hops max, 12 byte
packets
 1  2001:448::2d0:58ff:fef3:6d41  104.200 ms  106.136 ms  102.848 ms
 2  2001:468:43f:3::1  103.322 ms  104.173 ms  103.114 ms
 3  2001:18e8:ff00:3::2  102.911 ms  114.091 ms  108.443 ms
 4  2001:468:ff:144::1  158.249 ms  129.354 ms  134.121 ms
 5  2001:468:ff:109::2  171.069 ms  159.527 ms  145.838 ms
 6  abilene-wash.rt1.fra.de.geant2.net  396.144 ms  395.952 ms  409.959 ms
 7  2001:798:14:10aa::22  419.956 ms  414.580 ms  413.593 ms
 8  te-5-4.gate-2.bce.restena.lu  419.201 ms  414.804 ms  420.396 ms
 9  gate-1-v26.rest.restena.lu  424.745 ms  424.390 ms  411.707 ms
10  fw.restena.lu  430.896 ms  411.070 ms  410.803 ms
11  fw.restena.lu  411.747 ms !P  411.189 ms !P  413.056 ms !P
```

Tabla A.30 Traza de los paquetes utilizando el relay 6to4 en FreeBSD.

Linux

Como se mencionó anteriormente la forma más adecuada para crear túneles en Linux es por medio del comando *ip* como se muestra en la tabla A.31 donde se crea, levanta y configura el mecanismo de 6to4 a través del relay de la UNAM.

```
# ip tunnel add tun6to4 mode sit ttl 64 remote 132.248.108.254 local 132.248.10.24
# ip link set dev tun6to4 up
# ip address add 2002:84f8:6cf0::1/16 dev tun6to4
# route -A inet6 add default gw 2002: 84f8:6cfe::1
```

Tabla A.31 Configuración de 6to4 en Linux a través de un relay de la UNAM.

En la tabla A.32 se muestran las interfases del sistema después de la configuración del túnel mediante el relay 6to4 de la UNAM.

```
# /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:13:20:74:12:3c
          inet addr:132.248.10.24  Bcast:132.248.108.255  Mask:255.255.255.224
          inet6 addr: 2001:1218:1:6:213:20ff:fe74:123c/64 Scope:Global
          inet6 addr: fe80::213:20ff:fe74:123c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:494 errors:0 dropped:0 overruns:0 frame:0
          TX packets:595 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:419959 (410.1 KiB)  TX bytes:77351 (75.5 KiB)
          Interrupt:17
tun6to4   Link encap:IPv6-in-IPv4
          inet6 addr: 2002:84f8:6cf0::1/16 Scope:Global
          inet6 addr: fe80::84f8:6cf0/128 Scope:Link
          UP POINTOPOINT RUNNING NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:175 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:16760 (16.3 KiB)
```

Tabla A.32 Interfases en Linux después de la configuración de 6to4.

Verificación de la conectividad IPv6

Al término de la configuración se pueden realizar pruebas de conectividad a diversos sitios IPv6 como se muestra en la tabla A.33.

```
# ping6 www.ipv6forum.com
PING 2001:a18:1:20::22 (2001:a18:1:20::22) 56 data bytes
64 bytes from 2001:a18:1:20::22: icmp_seq=1 ttl=64 time=0.481 ms
64 bytes from 2001:a18:1:20::22: icmp_seq=2 ttl=64 time=0.455 ms
64 bytes from 2001:a18:1:20::22: icmp_seq=3 ttl=64 time=0.465 ms

--- 2001:448:11:1::1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.455/0.467/0.481/0.010 ms
```

Tabla A.33 Prueba de conectividad a través de un relay 6to4 de la UNAM en Linux.

C) En forma automática mediante Teredo.

Como se mencionó en el capitulo 3 la idea de teredo es poder establecer un túnel IPv6 en IPv4 a través de un Traductor de Direcciones de Red, NAT por sus siglas en inglés, en aquellos hosts con direcciones IPv4 no homologadas logrando la conectividad a través de estos dispositivos; a diferencia de 6to4 y túneles manuales donde los hosts tienen que utilizar direcciones IPv4 homologadas.

Windows XP

En los sistemas Windows se puede configurar el cliente Teredo para poder tener conectividad IPv6 cuando los hosts se encuentran detrás de un NAT, para esto se ejecutan los comandos que se muestran en la tabla A.34.

```
C:\>netsh
netsh>interface ipv6
netsh interface ipv6> set teredo client 132.24.10.24  60
```

Tabla A.34 Ejemplo de configuración en Windows XP de un cliente teredo a través de un servidor teredo de la UNAM.

Con esta configuración la dirección IPv4 del servidor especificada, se asigna una dirección IPv6 al cliente, el cual con la ayuda de los relays realiza el reenvío de los paquetes y así establecer conectividad a los sitios IPv6 como se muestra en las tablas A.35 y A.36.

Verificación de la conectividad IPv6

```
C:\>ping -6 www.ipv6forum.com
Haciendo ping a www.ipv6forum.com [2001:a18:1:20::22] con 32 bytes de datos:
Respuesta desde 2001:a18:1:20::22: tiempo=741ms
Respuesta desde 2001:a18:1:20::22: tiempo=247ms
Respuesta desde 2001:a18:1:20::22: tiempo=246ms
Respuesta desde 2001:a18:1:20::22: tiempo=247ms
Estadísticas de ping para 2001:a18:1:20::22:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0
   (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 246ms, Máximo = 741ms, Media = 370ms
```

Tabla A.35 Prueba de conectividad en Windows XP mediante un servidor teredo de la UNAM.

```
C:\>tracert -6 www.ipv6forum.com
Traza a la dirección www.ipv6forum.com [2001:a18:1:20::22]
sobre un máximo de 30 saltos:
  1   227 ms   227 ms   223 ms  teredo-relay.ipv6.lrz-muenchen.de [2001:4ca0:0:01:0:3544:1:1]
  2   236 ms   229 ms   228 ms  vl-60.csr1-2wr.lrz-muenchen.de [2001:4ca0:0:101:1]
  3   241 ms     *      226 ms  xr-gar1-te1-3-108.x-win.dfn.de [2001:638:c:a003:1]
  4   232 ms   233 ms   232 ms  zr-fra1-te0-7-0-1.x-win.dfn.de [2001:638:c:c043:2]
  5   467 ms   232 ms   266 ms  dfn.rt1.fra.de.geant2.net [2001:798:14:10aa::1]
  6   253 ms   247 ms   246 ms  2001:798:14:10aa::22
  7   247 ms   247 ms   247 ms  te-5-4.gate-2.bce.restena.lu [2001:a18:0:102::2
  8   252 ms   248 ms   246 ms  gate-1-v26.rest.restena.lu [2001:a18:0:ff01::1]
  9   253 ms   247 ms   255 ms  fw.restena.lu [2001:a18:0:408::4]
 10   246 ms   248 ms   248 ms  www.ipv6forum.org [2001:a18:1:20::22]
Traza completa.
```

Tabla A.36 Traza de los paquetes en Windows XP utilizando servidor teredo de la UNAM.

Windows Vista.

El cliente teredo en Windows Vista se encuentra habilitado por defecto, cuando se conecta un host que cuenta con Windows Vista a un dispositivo NAT, el host configurará una dirección IPv6 automáticamente a la interfaz Teredo con los parámetros que el sistema tiene configurados por defecto.

Para que los hosts puedan obtener una dirección IPv6 a través de un servidor teredo de la UNAM se ejecutan los comandos que se muestran en la tabla A.37 con la ayuda de netsh.

```
C:\>netsh
netsh>interface ipv6
netsh interface ipv6> set teredo client 132.24.10.24  60
```

Tabla A.37 Ejemplo de configuración en Windows Vista del cliente teredo a través de un servidor teredo de la UNAM.

En la tabla A.38 se muestran las interfases después de la configuración del cliente Teredo en Windows Vista.

```
C:\>ipconfig
Configuración IP de Windows
Adaptador de Ethernet Conexión de área local:
  Sufijo DNS específico para la conexión. . : undesirable-ipv6.netlab.unam.mx
  Vínculo: dirección IPv6 local. . . : fe80::f855:966d:7507:121b%8
  Dirección IPv4. . . . . . . . . . . . . : 192.168.108.51
  Máscara de subred . . . . . . . . . . . : 255.255.255.0
  Puerta de enlace predeterminada . . . . . : 192.168.108.254
Adaptador de túnel Conexión de área local*:
  Estado de los medios. . . . . . . . . . . : medios desconectados
  Sufijo DNS específico para la conexión. . : undesirable-ipv6.netlab.unam.mx
Adaptador de túnel Conexión de área local* 9:
  Sufijo DNS específico para la conexión. . :
  Dirección IPv6 . . . . . . . . . . . : 2001:0:53aa:64c:24f4:78f8:7b07:9315
  Vínculo: dirección IPv6 local. . . : fe80::24f4:78f8:7b07:9315%10
  Puerta de enlace predeterminada . . . . . : ::
```

Tabla A.38 Configuración de las interfases de Windows Vista después de habilitar cliente teredo de la UNAM.

Verificación de la conectividad IPv6

Después de la configuración del cliente Teredo se pueden realizar pruebas de conectividad con los comandos ping y tracert a sitios IPv6, un ejemplo se muestra en las tablas A.39 y A.40.

```
C:\> ping -6 www.ipv6forum.com
Haciendo ping a www.ipv6forum.com [2001:a18:1:20::22] desde 2001:0:53aa:64c:24f4
:78f8:7b07:9315 con 32 bytes de datos:
Respuesta desde 2001:a18:1:20::22: tiempo=477ms
Respuesta desde 2001:a18:1:20::22: tiempo=247ms
Respuesta desde 2001:a18:1:20::22: tiempo=248ms
Estadísticas de ping para 2001:a18:1:20::22:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 247ms, Máximo = 477ms, Media = 314ms
```

Tabla A.39 Prueba de conectividad en Windows Vista mediante un servidor teredo de la UNAM.

```
C:\> tracert -6 www.ipv6forum.com
Traza a la dirección www.ipv6forum.com [2001:a18:1:20::22]
sobre un máximo de 30 saltos:
  1   227 ms   227 ms   223 ms  teredo-relay.ipv6.lrz-muenchen.de [2001:4ca0:0:01:0:3544:1:1]
  2   236 ms   229 ms   228 ms  vl-60.csr1-2wr.lrz-muenchen.de [2001:4ca0:0:101:1]
  3   241 ms     *      226 ms  xr-gar1-te1-3-108.x-win.dfn.de [2001:638:c:a003:1]
  4   232 ms   233 ms   232 ms  zr-fra1-te0-7-0-1.x-win.dfn.de [2001:638:c:c043:2]
  5   467 ms   232 ms   266 ms  dfn.rt1.fra.de.geant2.net [2001:798:14:10aa::1]
  6   253 ms   247 ms   246 ms  2001:798:14:10aa::22
  7   247 ms   247 ms   247 ms  te-5-4.gate-2.bce.restena.lu [2001:a18:0:102::2
  8   252 ms   248 ms   246 ms  gate-1-v26.rest.restena.lu [2001:a18:0:ff01::1]
  9   253 ms   247 ms   255 ms  fw.restena.lu [2001:a18:0:408::4]
 10   246 ms   248 ms   248 ms  www.ipv6forum.org [2001:a18:1:20::22]
Traza completa.
```

Tabla A.40 Traza de los paquetes en Windows Vista utilizando servidor teredo de la UNAM.

Linux

Existe una implementación de Teredo para Linux llamada miredo, es una aplicación de software libre que puede realizar túneles IPv6 teredo e incluye todos los componentes que se especifican en este mecanismo como son cliente, relay y servidor teredo, para así obtener conectividad IPv6 cuando se está detrás de un dispositivo NAT en los sistemas Linux.

Para hacer uso del software miredo es necesario descargarlo de la página http://www.remlab.net/files/miredo/ e instalar el software como se mencionó en el capitulo 8. Una vez instalado, se configura el archivo /etc/miredo.conf que habilitará el host como cliente miredo como se muestra en la tabla A.41.

```
netlab@netlab:/etc$ more miredo.conf
RelayType restricted
# Please refer to the miredo.conf(5) man page for details.
InterfaceName    teredo
# Pick a Teredo server:
#ServerAddress    teredo-debian.remlab.net
ServerAddress 132.24.10.24
# Some firewall/NAT setups require a specific UDP port number:
#BindPort    3545
```

Tabla A.41 Ejemplo de configuración del cliente teredo en Linux mediante el software miredo.

Al término de la configuración se inicia el programa con la siguiente instrucción:
/usr/local/sbin/miredo
Realizado este paso se levanta automáticamente una nueva interfaz en el sistema como se muestra en la tabla A.42.

```
# ifconifg
ath0      Link encap:Ethernet  dirección HW 00:1f:e1:37:35:c2
          inet dirección:10.4.250.242  Difusión:10.4.250.255  MÃ¡scara:255.255.255.0
          dirección inet6: fe80::21f:e1ff:fe37:35c2/64 Alcance:VÃnculo
          ARRIBA DIFUSIÃ“N CORRIENDO MULTICAST  MTU:1500  MÃ©trica:1
          RX packets:1870 errors:0 dropped:0 overruns:0 frame:0
          TX packets:970 errors:2 dropped:2 overruns:0 carrier:0
          RX bytes:1494154 (1.4 MB)  TX bytes:129525 (126.4 KB)

teredo    Link encap:UNSPEC  direcciónHW 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          dirección inet6: fe80::ffff:ffff:ffff/64 Alcance:Vinculo
          dirección inet6: 2001:0:84f8:6cf0:8c4:6caa:7b08:a1e/32 Alcance:Global
          ARRIBA PUNTO A PUNTO CORRIENDO NOARP  MTU:1280  MÃ©trica:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
          colisiones:0 txqueuelen:500
          RX bytes:416 (416.0 B)  TX bytes:816 (816.0 B)
```

Tabla A.42 Interfaz teredo creada después de la configuración del software Miredo.

Verificación de la conectividad IPv6

Realizada la configuración del mecanismo se pueden realizar pruebas de conectividad IPv6 a través de Teredo con los comandos ping6 y traceroute6 como se muestra en las tablas A.43 y A.44.

```
# ping6 www.ipv6forum.com
PING www.ipv6forum.com(www.ipv6forum.org) 56 data bytes
64 bytes from www.ipv6forum.org: icmp_seq=1 ttl=55 time=880 ms
64 bytes from www.ipv6forum.org: icmp_seq=2 ttl=55 time=249 ms
64 bytes from www.ipv6forum.org: icmp_seq=3 ttl=55 time=263 ms
64 bytes from www.ipv6forum.org: icmp_seq=4 ttl=55 time=245 ms
--- www.ipv6forum.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 245.337/409.623/880.472/271.925 ms
```

Tabla A.43 Prueba de conectividad en Linux mediante un servidor teredo de la UNAM.

```
# traceroute6 www.ipv6forum.com
traceroute to www.ipv6forum.com (2001:a18:1:20::22) from 2001:0:84f8:6cf0:8c4:6caa:7b08:a1e, 30 hops
max, 16 byte packets
 1  teredo-relay.ipv6.lrz-muenchen.de (2001:4ca0:0:101:0:3544:1:1)  1254.34 ms  222.479 ms  224.789 ms
 2  vl-60.csr1-2wr.lrz-muenchen.de (2001:4ca0:0:101::1)  621.59 ms  222.565 ms  226.548 ms
 3  xr-gar1-te1-3-108.x-win.dfn.de (2001:638:c:a003::1)  619.784 ms  223.043 ms  224.249 ms
 4  zr-fra1-te0-7-0-1.x-win.dfn.de (2001:638:c:c043::2)  627.146 ms  231.527 ms  232.698 ms
 5  dfn.rt1.fra.de.geant2.net (2001:798:14:10aa::1)  626.621 ms  232.174 ms  233.787 ms
 6  restena-gw.rt1.fra.de.geant2.net (2001:798:14:10aa::22)  658.224 ms  254.648 ms  245.592 ms
 7  te-5-4.gate-2.bce.restena.lu (2001:a18:0:102::2)  656.752 ms  246.432 ms  254.082 ms
 8  gate-1-v26.rest.restena.lu (2001:a18:0:ff01::1)  661.924 ms  248.68 ms  251.807 ms
 9  fw.restena.lu (2001:a18:0:408::4)  657.517 ms  245.446 ms  246.209 ms
10  fw.restena.lu (2001:a18:0:408::4)  248.202 ms !S  252.571 ms !S  244.928 ms !S
```

Tabla A.44 Traza de los paquetes en Linux utilizando un servidor teredo de la UNAM.

ANEXO 2

I. Descripción sobre uso del servidor Túnel Broker de la UNAM para obtener conectividad IPv6.

El servidor de túneles broker permite a los usuarios de RedUNAM acceder a una página web con el fin de registrarse y autenticarse en el sistema para descargar un script de comandos mediante el cual podrán crear un túnel IPv6 sobre IPv4 y obtener conectividad IPv6 al Internet por medio de esta conexión punto a punto.

Para los usuarios de RedUNAM que deseen hacer uso de este servicio es necesario que accedan al Servidor de Túnel Broker a través de la dirección: http://tunelbroker.ipv6.unam.mx:8080, cuya página principal se muestra en la Figura B.1.

Figura B.1 Página principal del servidor del Túnel Broker.

Una vez en la página los usuarios deben registrarse para hacer uso del servicio que ofrece este servidor dando click a la liga de “REGISTRO” que se encuentra en la parte inferior de la página y proporcionando los datos que se solicitan como se muestra en la figura B.2 y con éstos se pueda llevar a cabo el registro correspondiente.

Figura B.2 Página de registro de usuarios.

Es importante recalcar que este servicio de conectividad IPv6 es solamente para usuarios que se encuentran dentro de RedUNAM ya que se busca que la universidad pueda ofrecer más alternativas, generadas dentro de la misma, para la transición a IPv6, es por eso que si se intenta registrar al servicio un usuario que se encuentre fuera de RedUNAM, entonces se mostrará la página de la figura B.3 y no se le brindará el servicio.

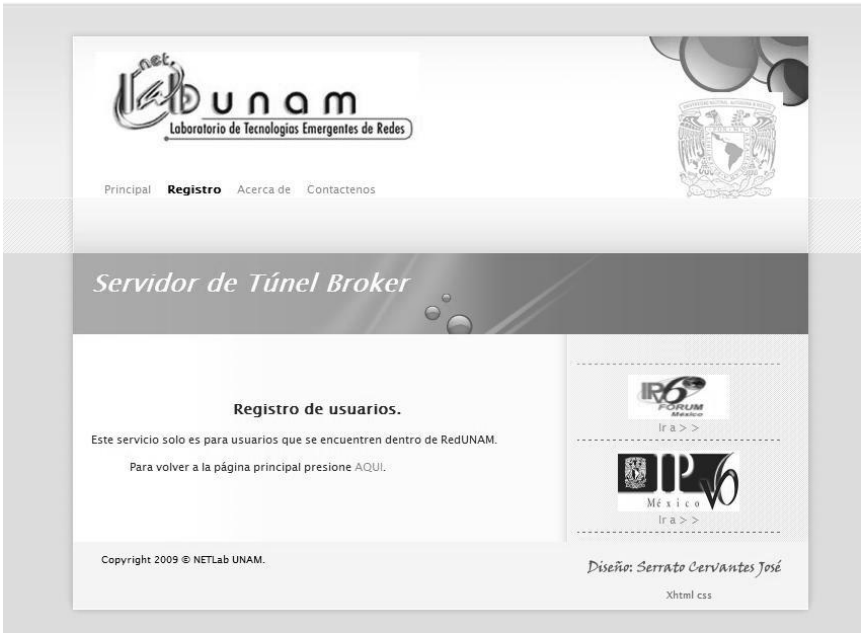


Figura B.3 Página de aviso para usuarios externos a RedUNAM.

Existe también el caso particular de los usuarios de RedUNAM que se encuentran conectados mediante la Red Inalámbrica Universitaria, RIU, que no cumpliría con el requerimiento de que el cliente tenga una IPv4 homologada para realizar el túnel punto a punto, dado que están detrás de un NAT, y como este mecanismo no es el adecuado para ofrecerles conectividad IPv6 a estos usuarios, al momento de su registro, se les presentará una página como la figura B.4 donde se les indicará que no pueden hacer uso de este servicio.

Figura B.4 Página de aviso para usuarios que realizan su conexión a través de la RIU.

El objetivo del registro, es llevar un control de los usuarios, que vayan a utilizar el servidor de túneles, a través de un sistema de autenticación como se muestra en la figura B.1 de la página principal. Así es como se permite que los usuarios que han realizado el registro puedan empezar con el proceso para obtener conectividad IPv6.

Una vez que el usuario se ha autenticado en el sistema, se le muestra una página de bienvenida como la de la figura B.5 en donde encuentra una pequeña descripción del servicio.

Figura B.5 Página de bienvenida del Servidor de Túneles Broker.

Continuando con el proceso, el sistema direcciona a la página de administración de conexiones por túnel que se muestra en la figura B.6 donde, también se presenta en forma de tabla los túneles que se han configurado y un pequeño resumen en la parte derecha con los datos generales del usuario.

Figura B.6 Página de administración de conexiones por túnel.

En la parte inferior de la página se encuentra la liga "CREAR UN TÚNEL" la cual direcciona a la página donde se encuentran los datos necesarios para crear el túnel IPv6 sobre IPv4 y con ellos se generarán los scripts de configuración como se muestra en la figura B.7.

Figura B.7 Página de creación de túneles.

Como se observa en la figura B.7 los datos del nombre de túnel y la dirección IPv4 se llenan automáticamente por el sistema lo que solo deja al usuario la selección del sistema operativo del host en el cual se llevará a cabo la configuración del túnel.

Los sistemas operativos que se listan son:

- Windows XP
- Windows Vista
- Sistemas Linux.

Al realizar la selección y presionar el botón de "CREAR TÚNEL" el sistema devuelve una página de confirmación, donde se presentará si el túnel se creo o no satisfactoriamente, como se muestra en la figura B.8.

Figura B.8 Pantalla de confirmación de creación del túnel.

Al confirmar que el túnel se ha creado de forma satisfactoria del lado del servidor, se debe presionar en la liga que lleva a la página de administración de túneles en donde se ven reflejados los cambios que se acaban de realizar como se muestra en la figura B.9.

Figura B.9 Página de administración conexiones por túnel.

De acuerdo a la figura anterior se agrega una entrada en la tabla que presenta los datos más importantes del túnel que se dio de alta hasta el momento sólo del lado del servidor, los cuales son:

- Nombre del túnel. Es el nombre de la interfaz que se creará en el equipo del usuario con el script de configuración.
- IPv4 Local. Es la IPv4 del host hasta donde llegará el túnel.
- IPv6 Local. Es la IPv6 que se le ha asignado al host que creará el túnel.
- IPv6 remota. Es la IPv6 del servidor de túneles que se le conoce como el punto final del túnel.
- Script de conexión. Es la liga a la página donde se podrán descargar los scripts de configuración o ver los comandos para configurar el túnel de forma manual.
- Borrar túnel. Es la liga para borrar el túnel existente.

Se puede ver en la figura B.9 que la liga de "CREAR UN TÚNEL" ya no se encuentra, debido a que este servidor de túneles sólo permite realizar un túnel hacia un sólo equipo por usuario y sólo a una dirección IPv4, por lo que para realizar otro túnel es necesario borrar el existente.

Como se mencionó, este servidor de túneles puede proporcionar conectividad IPv6 a clientes con diversos sistemas operativos, dependiendo la selección realizada en la página de ***creación de túneles,*** por lo que a continuación se describe el procedimiento a seguir en cada sistema operativo.

Windows Vista

Como se mencionó anteriormente, hasta el momento ya se habrá dado de alta un túnel del lado del servidor, por lo que sólo resta configurar el túnel del lado del cliente, desde la página de A*dministración de conexiones por túnel* figura B.9, al presionar el enlace de la columna *Script de conexión*, el sistema direcciona a la página de *descarga de script de configuración* como se muestra en la figura B.10.

Figura B.10 Página de descarga del script de configuración para Windows Vista.

En la parte superior se encuentran las ligas de "*descargar script para activar automáticamente el túnel IPv6/IPv4"* y de "*descargar script para desactivar automáticamente el túnel IPv6/IPv4"* con las cuales se pueden descargar los scripts de configuración automática como se muestra en la figura B.11 y adicionalmente se muestran los comandos para poder configurar el túnel de forma manual.

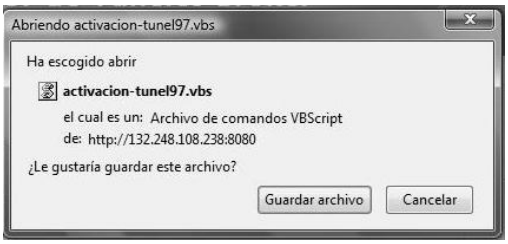


Figura B.11 Ventana de descarga de script de configuración.

Una vez que se han descargado los scripts de configuración, estos se deben ejecutar con privilegios de administrador para que puedan realizar los cambios requeridos para levantar el túnel o borrar el mismo y obtener conectividad IPv6.
Después que se ejecute el script de configuración, las interfases en Windows Vista mostrarán algo similar al ejemplo de la tabla B.1.

```
C:\ >ipconfig
Configuración IP de Windows
Adaptador LAN inalámbrico Conexión de red inalámbrica:
  Sufijo DNS específico para la conexión. . :
  Vínculo: dirección IPv6 local. . . : fe80::d087:c07a:716e:50cf%9
  Dirección IPv4. . . . . . . . . . . . . : 132.248.214.94
  Máscara de subred . . . . . . . . . . . . : 255.255.255.192
  Puerta de enlace predeterminada . . . . . : 132.248.214.126
Adaptador de túnel DGSCA72:
  Sufijo DNS específico para la conexión. . :
  Dirección IPv6 . . . . . . . . . . . : 2001:448:20:a55a::7090
  Puerta de enlace predeterminada . . . . . : 2001:448:20:a55a::61
```

Tabla B.1 Ejemplo de interfases de Windows Vista después de la ejecución del script de configuración.

Al verificar la configuración de la interfaz del túnel, se puede comprobar la conectividad a un sitio IPv6, visitando páginas Web con soporte IPv6 como las que se acceden dando click a las 3 imágenes que aparecen al final de la página de la figura B.10, y/o con los comandos ping y tracert como se muestra en las tablas B.2 y B.3.

```
C:\>ping www.ipv6forum.com
Haciendo ping a www.ipv6forum.com [2001:a18:1:20::22] desde 2001:448:20:a55a::7090 con 32 bytes de datos:
Respuesta desde 2001:a18:1:20::22: tiempo=611ms
Respuesta desde 2001:a18:1:20::22: tiempo=618ms
Estadísticas de ping para 2001:a18:1:20::22:
   Paquetes: enviados = 2, recibidos = 2, perdidos = 0
   (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 611ms, Máximo = 630ms, Media = 622ms
```

Tabla B.2 Prueba de conectividad por medio del servidor de túneles broker.

```
C:\>tracert www.ipv6forum.com
Traza a la dirección www.ipv6forum.com [2001:a18:1:20::22]
sobre un máximo de 30 saltos:
  1   172 ms   187 ms   202 ms  2001:448:20:a55a::61
  2   296 ms     *      203 ms  2001:448::2d0:58ff:fef3:6d41
  3   218 ms   202 ms   187 ms  2001:1228:11b:90a::1
  4   296 ms   233 ms   218 ms  2001:1228:10a:f09::1
  5   251 ms   249 ms   249 ms  2001:1228:10a:f02::2
  6   343 ms   215 ms   246 ms  2001:1348:1:2::1
  7   437 ms   327 ms   343 ms  2001:1348::12
  8   437 ms   452 ms   483 ms  2001:1348::1a
  9   406 ms   468 ms   513 ms  2001:1348::1e
 10   451 ms   450 ms   453 ms  2001:1348::26
 11   667 ms   652 ms   702 ms  clara.rt1.mad.es.geant2.net [2001:798:17:10aa::9]
 12   827 ms   667 ms   937 ms  so-7-2-0.rt1.gen.ch.geant2.net [2001:798:cc:1201:1701::1]
 13     *      545 ms   531 ms  so-3-3-0.rt1.fra.de.geant2.net [2001:798:cc:1201:1401::6]
 14   562 ms   562 ms   546 ms  restena-gw.rt1.fra.de.geant2.net [2001:798:14:10aa::22]
 15   544 ms   562 ms   561 ms  te-5-4.gate-2.bce.restena.lu [2001:a18:0:102::2]
 16   562 ms   531 ms   562 ms  gate-1-v26.rest.restena.lu [2001:a18:0:ff01::1]
 17   546 ms   546 ms   541 ms  fw.restena.lu [2001:a18:0:408::4]
 18   559 ms   637 ms   546 ms  www.ipv6forum.org [2001:a18:1:20::22]
Traza completa.
```

Tabla B.3 Traza de los paquetes pasando por el servidor de túneles broker.

Windows XP

Si al crear el túnel se seleccionó el sistema operativo Windows XP entonces el resultado de la página de administración de conexiones por túnel es como el que se muestra en la figura B.9, al presionar el enlace de la columna *Script de conexión*, el sistema direcciona a la página de *descarga de script de conexión* como se muestra en la figura B.12.

Figura B.12 Página de descarga de script de conexión para Windows XP.

En la figura anterior se muestran instrucciones que son necesarias para que se pueda establecer conectividad IPv6 en los sistemas Windows XP y que pueden ser ejecutadas por los usuarios, así como los enlaces de "*descargar script para activar automáticamente el túnel IPv6/IPv4"* y "*descargar script para desactivar automáticamente el túnel IPv6/IPv4"* con los cuales se obtienen los scripts de configuración automática como se mostró en la figura B.11. Adicionalmente se muestran los comandos para poder levantar el túnel de forma manual.
Los scripts que se descargan del servidor deben ser ejecutados con privilegios de administrador para que puedan realizar los cambios necesarios para establecer conectividad IPv6 a través del túnel broker. Al término de la ejecución del script las interfases en Windows XP mostrarán algo similar al ejemplo de la tabla B.4.

```
c:\>ipconfig
Adaptador Ethernet Conexión de área local        :
      Sufijo de conexión específica DNS :
      Dirección IP. . . . . . . . . . . : 132.248.214.94
      Máscara de subred . . . . . . . . : 255.255.255.192
      Dirección IP. . . . . . . . . . . : fe80::211:11ff:fe2b:40f2%5
      Puerta de enlace predeterminada   : 132.248.214.126
Configuración IP de Windows
      Sufijo DNS principal  . . . . . . :
      Tipo de nodo . . . . . . . . . .  : desconocido
      Enrutamiento habilitado. . . . . .: No
      Proxy WINS habilitado. . . . .    : No
Adaptador de túnel DGSCA221          :
      Sufijo de conexión específica DNS :
      DHCP habilitado. . . . . . . . .  : No
      Dirección IP. . . . . . . . . . . : 2001:448:20:56cf::48f3
      Dirección IP. . . . . . . . . . . : fe80::8:84f8:7157%8
      Puerta de enlace predeterminada   : 2001:448:20: 56cf::64
      Servidores DNS . . . . . . . . . .: fec0:0:0:ffff::1%6
                                  fec0:0:0:ffff::2%6
      NetBios sobre TCPIP. . . . . . .  : Deshabilitado
```

Tabla B.4 Interfases de Windows XP después de la ejecución del script de configuración.

Concluida la configuración se puede comprobar la conectividad a sitios IPv6 visitando páginas Web con soporte IPv6 a las que se acceden presionando sobre cualquiera de las tres imágenes que aparecen al final de la página de la figura B.12, y/o con los comandos ping y tracert, así se tendrá la seguridad de poder acceder a sitios IPv6 mediante el túnel como se muestra en las tablas B.5 y B.6.

```
C:\>ping www.ipv6forum.com
Haciendo ping a www.ipv6forum.com [2001:a18:1:20::22] desde 2001:448:20:56cf::48f3 con 32 bytes de
datos:
Respuesta desde 2001:a18:1:20::22: tiempo=611ms
Respuesta desde 2001:a18:1:20::22: tiempo=620ms
Respuesta desde 2001:a18:1:20::22: tiempo=629ms
Respuesta desde 2001:a18:1:20::22: tiempo=618ms
Estadísticas de ping para 2001:a18:1:20::22:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 611ms, Máximo = 629ms, Media = 624ms
```

Tabla B.5 Prueba de conectividad por medio del servidor de túneles broker.

```
c:\>tracert www.ipv6forum.com
Traza a la dirección www.ipv6forum.com [2001:a18:1:20::22]
sobre un máximo de 30 saltos:
  1   172 ms   187 ms   202 ms  2001:448:20:56cf::64
  2   296 ms     *      203 ms  2001:448::2d0:58ff:fef3:6d41
  3   218 ms   202 ms   187 ms  2001:1228:11b:90a::1
  4   296 ms   233 ms   218 ms  2001:1228:10a:f09::1
  5   251 ms   249 ms   249 ms  2001:1228:10a:f02::2
  6   343 ms   215 ms   246 ms  2001:1348:1:2::1
  7   437 ms   327 ms   343 ms  2001:1348::12
  8   437 ms   452 ms   483 ms  2001:1348::1a
  9   451 ms   450 ms   453 ms  2001:1348::26
 10   667 ms   652 ms   702 ms  clara.rt1.mad.es.geant2.net [2001:798:17:10aa::9]
 11   827 ms   667 ms   937 ms  so-7-2-0.rt1.gen.ch.geant2.net [2001:798:cc:1201:1701::1]
 12     *      545 ms   531 ms  so-3-3-0.rt1.fra.de.geant2.net [2001:798:cc:1201:1401::6]
 13   562 ms   562 ms   546 ms  restena-gw.rt1.fra.de.geant2.net [2001:798:14:10aa::22]
 14   544 ms   562 ms   561 ms  te-5-4.gate-2.bce.restena.lu [2001:a18:0:102::2]
 15   562 ms   531 ms   562 ms  gate-1-v26.rest.restena.lu [2001:a18:0:ff01::1]
 16   546 ms   546 ms   541 ms  fw.restena.lu [2001:a18:0:408::4]
 17     *      637 ms   546 ms  www.ipv6forum.org [2001:a18:1:20::22]
Traza completa.
```

Tabla B.6 Traza de los paquetes pasando por el servidor de túneles broker.

Linux

Si al crear el túnel se seleccionó el sistema operativo Linux el resultado de la página de *administración de conexiones por túnel* es como el que se muestra en la figura B.9, al presionar el enlace de la columna *Script de conexión*, el sistema direcciona a la página de *descarga de script de conexión* como se muestra en la figura B.13.

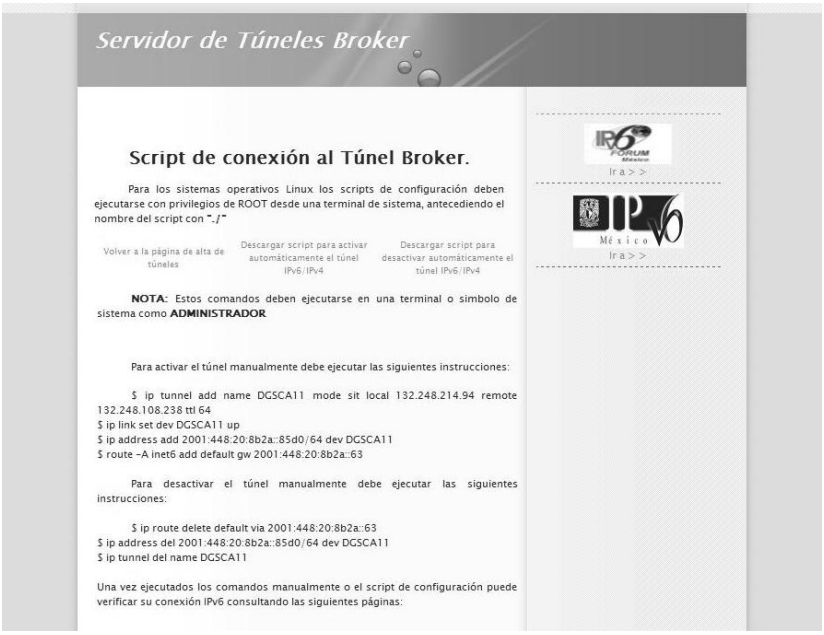


Figura B.13 Página descarga de script de configuración para Linux.

En la figura anterior se encuentran los enlaces de "*descargar script para activar automáticamente el túnel IPv6/IPv4"* y "*descargar script para desactivar automáticamente el túnel IPv6/IPv4"* con las cuales se pueden descargar los scripts de configuración automática, para los sistemas Linux, como se muestra en la figura B.11 y los comando para poder configurar el túnel de forma manual.

Después de descargar el script de configuración es necesario ejecutarlo como ROOT mediante una terminal de sistema, antecedido por "./"<nombre del script> para así tener el túnel configurado de forma automática y obtener conectividad IPv6. La configuración de las interfases después de ejecutar el script mostrará algo similar al ejemplo de la tabla B.7.

```
# /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:13:20:74:12:3c
          inet addr:132.248.214.94  Bcast:132.248.108.255  Mask:255.255.255.192                UP
BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:494 errors:0 dropped:0 overruns:0 frame:0
          TX packets:595 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:419959 (410.1 KiB)  TX bytes:77351 (75.5 KiB)
          Interrupt:17
DGSCA11   Link encap:IPv6-in-IPv4
          inet6 addr: 2001:448:20:8b2a::85d0/64 Scope:Global
          UP POINTOPOINT RUNNING NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:175 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:16760 (16.3 KiB )
```

Tabla B.7 Interfases en Linux después de la ejecución del script de configuración.

Finalizada la configuración se puede acceder a sitios IPv6 o realizar pruebas de conectividad con los comandos ping6 y traceroute6, como se muestra en las tablas B.8 y B.9.

```
# ping6 www.ipv6forum.com
PING www.ipv6forum.com(www.ipv6forum.org) 56 data bytes
64 bytes from www.ipv6forum.org: icmp_seq=1 ttl=55 time=880 ms
64 bytes from www.ipv6forum.org: icmp_seq=2 ttl=55 time=249 ms
64 bytes from www.ipv6forum.org: icmp_seq=3 ttl=55 time=263 ms
64 bytes from www.ipv6forum.org: icmp_seq=4 ttl=55 time=245 ms
--- www.ipv6forum.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 245.337/409.623/880.472/271.925 ms
```

Tabla B.8 Prueba de conectividad por medio del servidor de túneles broker.

```
# traceroute6 www.ipv6forum.com
traceroute to www.ipv6forum.com (2001:a18:1:20::22) from 2001:448:20:8b2a::85d0, 30 hops max,
16 byte packets
 1 2001:448:20:8b2a::63    1254.34 ms  222.479 ms  224.789 ms
 2 2001:448::2d0:58ff:fef3:6d41  621.59 ms  222.565 ms  226.548 ms
 3 2001:1228:11b:90a::1  619.784 ms  223.043 ms  224.249 ms
 4 2001:1228:10a:f09::1  627.146 ms  231.527 ms  232.698 ms
 5  dfn.rt1.fra.de.geant2.net (2001:798:14:10aa::1)  626.621 ms  232.174 ms  233.787 ms
 6  restena-gw.rt1.fra.de.geant2.net (2001:798:14:10aa::22)  658.224 ms  254.648 ms  245.592 ms
 7  te-5-4.gate-2.bce.restena.lu (2001:a18:0:102::2)  656.752 ms  246.432 ms  254.082 ms
 8  gate-1-v26.rest.restena.lu (2001:a18:0:ff01::1)  661.924 ms  248.68 ms  251.807 ms
 9  fw.restena.lu (2001:a18:0:408::4)  657.517 ms  245.446 ms  246.209 ms
10  fw.restena.lu (2001:a18:0:408::4)  248.202 ms !S  252.571 ms !S  244.928 ms !S
```

Tabla B.9 Traza de los paquetes pasando por el servidor de túneles broker.

ANEXO 3

I. Scripts desarrollados para realizar conexiones IPv6 de forma automática en RedUNAM .

En ocasiones existen tareas simples, las cuales resultan repetitivas, que al llevarlas a cabo requieren de tiempo que podría ser aprovechado para realizar otras actividades de mayor importancia o simplemente que el esfuerzo invertido sea más redituable, es por eso que la automatización de tareas debe ser considerada por los administradores de red para realizarlas con un mínimo esfuerzo.

Los scripts son programas muy simples, que pueden ser hechos en un simple editor de textos con unas cuantas instrucciones. Los scripts son archivos de texto plano que pueden ser escritos y ejecutados sin un ambiente de desarrollo o compilador, ya que éstos son interpretados, es decir, que otros programas se encargan de leer y ejecutar los comandos del script línea por línea. Es extraño decir que se utilizan procesadores de texto para hacer un script, pero el resultado se traduce en que éstos son ideales para realizar tareas rápidas y un tanto tediosas, como una operación para manejar un archivo, administración de conexiones de red, e incluso el inicio de varios programas con un solo click.

Microsoft soporta varias tecnologías que se utilizan para la realización de scripts, que no han sido muy difundidas pero que son de gran utilidad, estas son: archivos batch, el Windows Script Host y scripts en Windows PowerShell.
Para la realización del trabajo se decidió utilizar Windows Script Host ya que PowerShell presenta grandes limitaciones en la portabilidad de los scripts dado que no todas las versiones del sistema operativo Windows son capaces de interpretarlos.

Windows Script Host

Windows Script Host es más flexible y robusto que batch, además que ofrece una mejor interacción con los usuarios, los scripts pueden ser ejecutados sobre cualquier equipo que esté bajo Windows 98 o superior, lo que representa una gran ventaja sobre PowerShell que no se encuentra instalado en todas las versiones del sistema Windows, representando un problema al compartir los scripts a otros equipos. Windows Script Host está diseñado para trabajar en un ambiente Windows por lo que puede tomar ventaja de sus servicios, redes y acceso a registros.

Windows Script Host es el encargado de ejecutar los scripts, ya que como se había mencionado los scripts son interpretados, por lo que en teoría Windows Script Host es independiente del lenguaje, lo que significa que puede extenderse a soportar cualquier lenguaje moderno de scripts como Perl y Python. Aunque para nuestro caso y como se utiliza generalmente en la programación de scripts en Windows Script Host se utilizó VBScript para su escritura.

VBScript está basado en otro lenguaje de programación de Microsoft, Visual Basic, que a su vez este se basa en el lenguaje BASIC. El propósito del uso de VBScript es porque es un lenguaje fácil de aprender e implementar, facilita el acceso a las características necesarias como el

manejo de archivos, acceso a registros y es muy parecido a Visual Basic que es uno de lo lenguajes más utilizados hoy en día.

Incluso VBScript es compatible y puede usarse tal cual en Visual Basic 6, a excepción de algunas instrucciones que tienen su similar en Visual Basic 6, por lo demás son exactamente idénticos, así que aprender VBScript es ampliamente recomendado para los programadores que deseen comenzar el aprendizaje de Visual Basic.

Para fines de esta tesis se diseñaron una serie de scripts que ayudan a las tareas de configuración de conexión, monitoreo y estadísticas de red, para que los usuarios y administradores de red puedan obtener de una forma sencilla los datos de su interés y conectividad a IPv6.

Scripts de monitoreo, estadísticas y configuración de IPv6 desarrollados.

Algo esencial a verificar en un host es si éste cuenta con soporte y conectividad IPv6, por lo que se desarrolló un script para este propósito llamado "***Verificación de soporte y conectividad IPv6"*** con el cual se pueden verificar de forma sencilla estos datos, como se muestra en la figura C.1, se despliega un pequeño menú con las opciones correspondientes. Al final de este anexo se incluye el código fuente de este script.

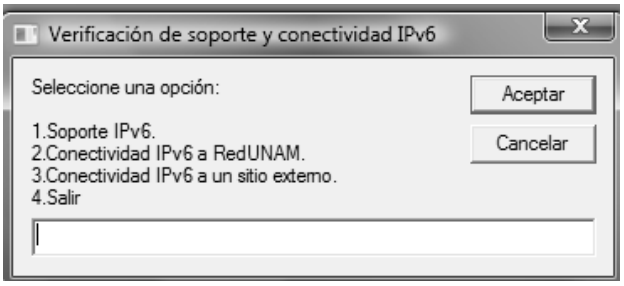


Figura C.1 Script de verificación de soporte y conectividad IPv6.

El script de *Verificación de soporte y conectividad IPv6* cuenta con tres opciones:

- Soporte IPv6. Con la cual se verifica si el equipo cuenta con el soporte IPv6 habilitado.
- Conectividad IPv6 a RedUNAM. Se verifica si el equipo puede establecer comunicación con el sitio IPv6 de la UNAM.
- Conectividad IPv6 a un sitio externo. Se verifica si el equipo puede establecer comunicación con un sitio IPv6 externo.

Después de haber realizado la configuración correspondiente para obtener conectividad IPv6 es posible verificar algunos parámetros, como es el caso de las direcciones IPv6 configuradas, la tabla de ruteo del equipo, etc. por lo que se desarrolló el script de ***"Configuración IPv6"*** como se muestra en la figura C.2.

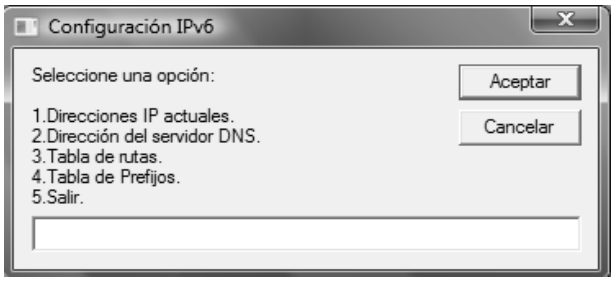


Figura C.2 Script de verificación de la configuración IPv6 actual.

El script de *Configuración IPv6* cuenta con las siguientes opciones:

- Direcciones IP actuales. Muestra un resumen con las interfases del sistema y las direcciones IPv6 asignadas a cada una de ellas.
- Dirección del servidor DNS. Muestra las direcciones del servidor DNS.
- Tabla de rutas. Muestra las entradas de la tabla de ruteo del host.
- Tabla de Prefijos. Muestra las entradas de la tabla de prefijos de sitios.

Los administradores en algunos casos deben tener conocimiento de datos referentes a estadísticas de IPv6, TCP, UDP, etc. Es por eso que el script de ***"Estadísticas IPv6"*** proporciona este tipo de datos como se muestra en la figura C.3.

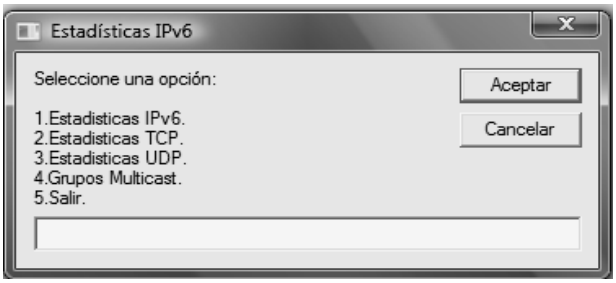


Figura C.3 Script de estadísticas IPv6 en un host.

El script de Estadísticas IPv6 cuenta con las siguientes opciones:

- Estadísticas IPv6. Muestras un resumen de las estadísticas sobre IPv6.
- Estadísticas TCP. Muestra un resumen de las estadísticas TCP.
- Estadísticas UDP. Muestra un resumen de las estadísticas UDP.
- Grupos Multicast. Muestra los grupos de multicast unidos.

Como se mencionó en el capitulo 3, se pueden realizar túneles IPv6 en IPv4, con lo cual se decidió realizar un script que facilitara la configuración manual de estos túneles entre dos equipos.
Así que el script ***"Configuración de túnel manual"*** se programó para este propósito realizando los siguientes pasos:

- Se ingresa la dirección IPv4 del equipo donde se configurará el túnel, como se muestra en al figura C.4.

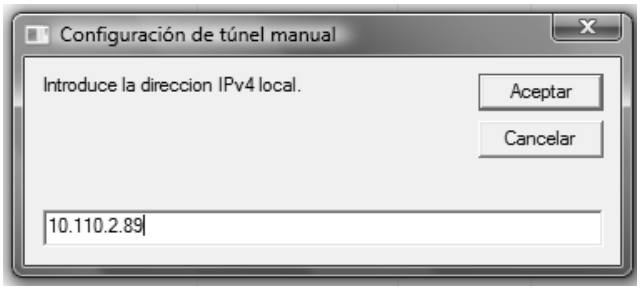


Figura C.4 Dirección IPv4 local para el extremo del túnel.

- Se ingresa la dirección IPv4 del equipo remoto donde se configurará el túnel, como se muestra en la figura C.5.

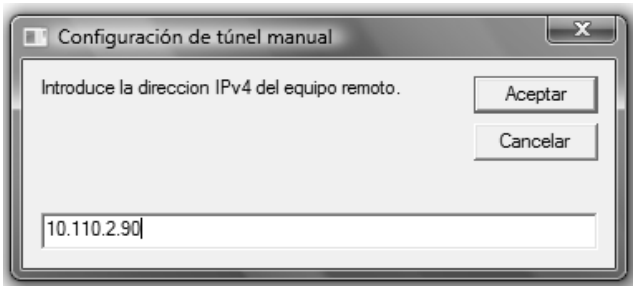


Figura C.5 Dirección IPv4 remota para el extremo del túnel.

- Se ingresa la dirección IPv6 local del túnel que tendrá nuestro equipo, como se muestra en la figura C.6.

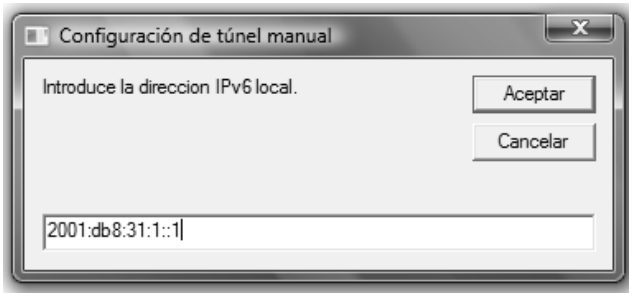


Figura C.6 Dirección IPv6 local para un extremo del túnel.

- Se ingresa la dirección IPv6 remota del túnel que será el extremo final del túnel, como se muestra en la figura C.7.

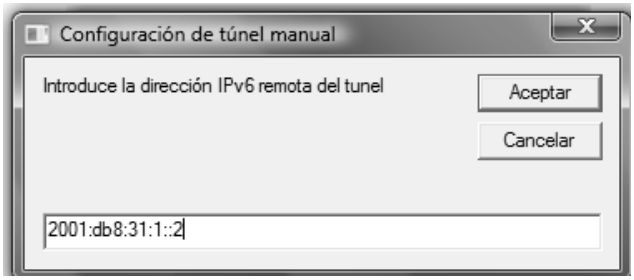


Figura C.7 Dirección IPv6 remota para el final del túnel.

- Con los datos capturados se muestra un resumen con todas las direcciones ingresadas para confirmar si éstas son las correctas como se muestra en la figura C.8

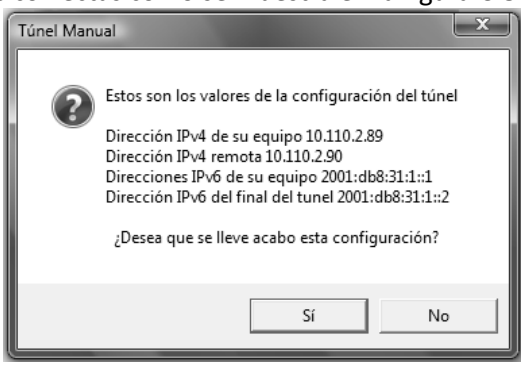


Figura C.8 Resumen de datos ingresados para la configuración del túnel.

- Después de confirmar los datos de la figura C.8 el script realiza la configuración del túnel y al término de la misma se despliega un mensaje de confirmación como el de la figura C.9.

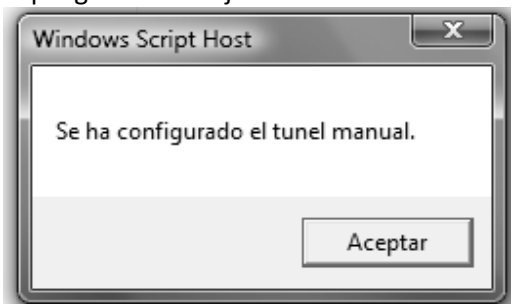


Figura C.9 Mensaje de confirmación de la configuración del túnel.

- Al momento de terminar la configuración se verifica si se puede establecer comunicación con el extremo del túnel configurado por el script y en caso de tener conectividad se mostrará un mensaje de confirmación, o en caso contrario se mostrará un mensaje como el de la figura C.10

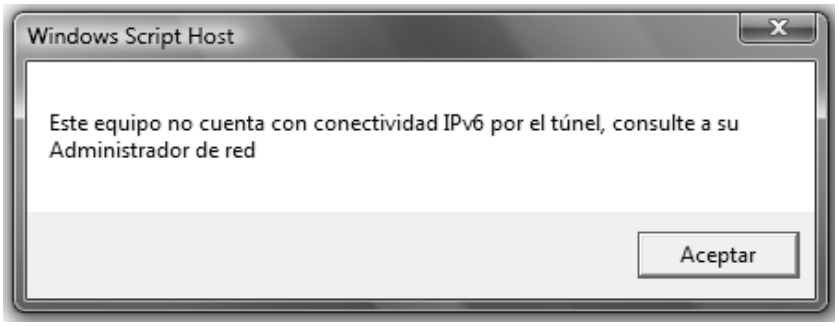


Figura C.10 Verificación de conectividad con el extremo del túnel.

En el host donde se ejecutó el script de configuración de túnel manual, se puede verificar el resultado de la configuración ejecutando en una ventana de comandos la instrucción **ipconfig** y observar los cambios realizados en las interfases por el script como se muestra en la tabla C.1.

```
C:\>ipconfig
Configuración IP de Windows
Adaptador LAN inalámbrico Conexión de red inalámbrica:
  Sufijo DNS específico para la conexión. . : unam.mx
  Vínculo: dirección IPv6 local. . . : fe80::d087:c07a:716e:50cf%9
  Dirección IPv4. . . . . . . . . . . . . : 10.110.2.89
  Máscara de subred . . . . . . . . . . . . : 255.255.255.0
  Puerta de enlace predeterminada . . . . . : 10.110.2.254
Adaptador de Ethernet Conexión de área local:
  Estado de los medios. . . . . . . . . . . : medios desconectados
  Sufijo DNS específico para la conexión. . :
Adaptador de túnel tun0:
  Sufijo DNS específico para la conexión. . : riu-fes.unam.mx
  Dirección IPv6 . . . . . . . . . . . : 2001:db8:31:1::1
  Vínculo: dirección IPv6 local. . . : fe80::c54d:c7fa:7c78:a6c6%28
  Puerta de enlace predeterminada . . . . . : 2001:db8:31:1::2
```

Tabla C.1 Interfases del host después de la configuración del túnel.

Después de verificar la interfaz creada por el script se puede realizar el mismo procedimiento en el equipo que será el otro extremo del túnel, y una vez configurado se puede realizar la prueba de conectividad entre ambos hosts, como se muestra en la tabla C.2.

```
C:\>ping 2001:db8:31:1::2
Haciendo ping a 2001:db8:31:1::2 desde 2001:db8:31:1::1 con 32 bytes de datos:
Respuesta desde 2001:db8:31:1::2: tiempo=1ms
Tiempo de espera agotado para esta solicitud.
Respuesta desde 2001:db8:31:1::2: tiempo=1ms
Respuesta desde 2001:db8:31:1::2: tiempo<1m
Estadísticas de ping para 2001:db8:31:1::2:
   Paquetes: enviados = 4, recibidos = 3, perdidos = 1
   (25% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 0ms, Máximo = 1ms, Media = 0ms
```

Tabla C.2 Ejemplo de prueba de conectividad entre hosts por medio del túnel.

En capítulos anteriores ya se han mencionado diferentes tipos de mecanismos de transición, su instalación, configuración e importancia, debido a esto se programó un script en el cual se puede tener acceso a los mecanismos que ofrece el NETLab para realizar la transición o empezar a usar IPv6 y con esto poner a disposición de los usuarios de RedUNAM una manera sencilla de obtener conectividad IPv6.

Se programó el script ***"Conectividad-IPv6_RedUNAM"*** en el cual se reúnen los mecanismos de transición implementados en este trabajo como se muestra en la Figura C.11.

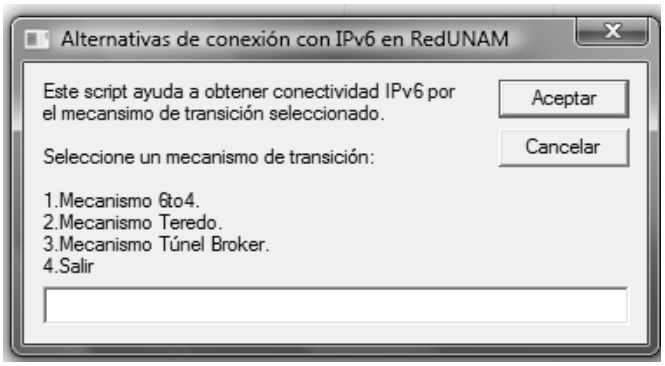


Figura C.11 Script de los diferentes mecanismos de transición a IPv6 ofrecidos por el NETLab.

En la figura anterior se pueden ver los tres mecanismos de transición a IPv6 que se encuentran funcionando en el NETLab:

- **6to4**
- **Teredo**
- **Túnel Broker**

Con los cuales se puede obtener conectividad IPv6, con la ayuda de este script los usuarios de RedUNAM pueden realizar la configuración de forma automática de cada uno de los mecanismos, con solo seleccionar la opción que se desee.

La primer opción que se puede seleccionar es el mecanismo **6to4,** al introducir la opción número uno se despliega un menú como el que se muestra en la figura C.12 donde se listan las opciones que puede realizar el script para este mecanismo.

Figura C.12 Script para la habilitación del mecanismo 6to4.

El script del mecanismo 6to4 cuenta con las siguientes opciones:

- Soporte IPv6: Verifica que el equipo cuente con el soporte IPv6.
- Verificación soporte 6to4: Verifica si el equipo cuenta con el soporte 6to4 habilitado.
- Habilitar 6to4 en Windows Vista: Habilita la conectividad IPv6 a través del mecanismo 6to4 para Windows Vista.
- Habilitar 6to4 en Windows XP: Habilita la conectividad IPv6 a través del mecanismo 6to4 para Windows XP.
- Deshabilitar 6to4: Deshabilita el soporte 6to4 en el equipo.
- Conectividad IPv6 en RedUNAM. Se verifica si el equipo puede establecer comunicación con el sitio IPv6 de la UNAM.
- Conectividad IPv6 a un sitio externo. Se verifica si el equipo puede establecer comunicación con un sitio IPv6 externo.

La segunda de las opciones que se puede seleccionar es el mecanismo de túnel **Teredo,** al introducir la opción número dos se despliega un menú como se muestra en la figura C.13 donde se listan las opciones que puede realizar el script para este mecanismo.

Figura C.13 Script para la habilitación del mecanismo Teredo.

- Soporte IPv6: Verifica que el equipo cuente con el soporte IPv6.
- Verificación soporte Teredo: Verifica si el equipo cuenta con el soporte Teredo habilitado.
- Habilitar Teredo: Habilita la conectividad IPv6 a través del mecanismo Teredo.
- Deshabilitar Teredo: Deshabilita el soporte Teredo en el equipo.
- Conectividad IPv6 en RedUNAM. Se verifica si el equipo puede establecer comunicación con el sitio IPv6 de la UNAM.
- Conectividad IPv6 a un sitio externo. Se verifica si el equipo puede establecer comunicación con un sitio IPv6 externo.

El último mecanismo que se puede seleccionar es el de **túnel Broker,** al introducir esta opción el script ejecutará en Internet Explorer la página del túnel Broker mencionada en el capitulo 8, como se muestra en la figura C.14, con lo cual los usuarios podrán realizar el registro correspondiente y así obtener conectividad IPv6 mediante este mecanismo.

Figura C.14 Página para obtener conectividad IPv6 a través de túnel Broker.

La implementación de los scripts fue sencilla debido a que VBScript es un lenguaje que cumplió con los requerimientos necesarios para su realización, como se puede ver en la tabla C.3 el código fuente de uno de los scripts, y es importante mencionar que los demás scripts fueron realizados de forma semejante para que cumplieran con los objetivos planteados en está tesis.

```
' VBScript:  Autor: José Guadalupe Serrato C.
' NOMBRE: < Verificación de soporte y conectividad IPv6.vbs>
'DEPENDENCIA: Laboratorio de Tecnologías Emergentes de Redes, NETLab, DGSCA-UNAM
'Tel: 5622 8857 o 56228526
'Email: staff@netlab.unam.mx
'Web: www.netlab.unam.mx
'El uso de este script está limitado solo con fines académicos y de investigación para usuarios de RedUNAM.
'Para cualquier otro fin se deberá notificar por escrito
'Hecho en México, Todos los Derechos reservados © 2009.
'Descripción:
'En este programa  hace  uso de la herramienta ping para verificar si nuestro equipo cuenta soporte y
conectividad IPv6.
'================================================================
Do until opcion = 4
opcion = InputBox ("Seleccione una opción:"&(Chr(13)& Chr(13))&"1.Soporte
IPv6."&(Chr(13))&"2.Conectividad IPv6 a RedUNAM."&(Chr(13))&"3.Conectividad IPv6 a un sitio
externo."&(Chr(13))&"4.Salir","Verificación de soporte y conectividad IPv6")

If opcion <> ""  Then
Select Case opcion
Case "1"
strcommand1 = "ping ::1"
subsoporte
Case "2"
strCommand = "ping -6 -n 3 2001:1218:1:6c::2"
subCommand
Case "3"
strCommand = "ping -6 www.ipv6forum.com"
subCommand
Case "4"
WScript.quit
Case Else
opcion = Asc (opcion)
MsgBox "La opción seleccionada es inválida",16 + 0,"Error"
End Select
Else
WScript.quit
End If
Loop

Sub subCommand
Set stdOut = WScript.StdOut
Set objShell = CreateObject("Wscript.shell")
Set objExec = objShell.exec ("Cmd /c " & strCommand)
strText = objExec.StdOut.readAll
resultado = Instr(1, strText,"Respuesta desde 2001:a18:1:20::22:")
resultado2 = Instr(1, strText,"Respuesta desde 2001:1218:1:6c::2:")
```

```
If resultado or resultado2 <> o Then
WScript.Echo "Este equipo cuenta con conectividad IPv6" & strText
Else
WScript.Echo "Este equipo no cuenta con conectividad IPv6"
End If
End Sub
Sub subsoporte
Set StdOut = WScript.StdOut
Set objShell = CreateObject("Wscript.shell")
Set objExec = objShell.exec ("Cmd /c " & strCommand1)
strText1 = objExec.StdOut.readAll
resultado1= Instr(1, strText1,"Respuesta desde ::1")
If resultado1 <> 0 Then
WScript.Echo "Este equipo cuenta con soporte IPv6 habilitado o instalado."
Else
WScript.Echo "Este equipo NO cuenta con soporte IPv6 habilitado o instalado."
End If
End Sub
```

Tabla C.3 Código fuente de uno de los scripts usando VBscript para este trabajo.