ESCUELA POLITECNICA DEL EJÉRCITO

FACULTAD DE SISTEMAS E INFORMÁTICA

IMPLEMENTACIÓN DE TUNNELING ENTRE REDES IPV4 E IPV6 PARA LA EMPRESA "NETXPERTS CONSULTING S.A."

PREVIA A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO EN SISTEMAS E INFORMÁTICA

POR: JÉSSICA PAOLA BARRERA ESPÍN

EDGAR MAURICIO GUERRA RODRÍGUEZ

SANGOLQUI, 13 DE JUNIO DEL 2005

ÍNDICE DE CONTENIDOS

LISTADO DE FIGURAS

LISTADO DE TABLAS

RESUMEN

El presente proyecto de tesis está encaminado a realizar la implementación de tunneling entre redes IPv6, utilizando para ello la infraestructura de conectividad IPv4 existente en la actualidad, para la empresa “netXperts Consulting S.A.”

La primera fase del proyecto comprende un estudio de todos los fundamentos teóricos necesarios para comprender la funcionalidad del protocolo IPv6 y su aplicación en el desarrollo de soluciones, tales como el tunneling entre redes. Posterior a este estudio se realizó la configuración del protocolo IPv6, tanto en computadores con sistemas operativos de las plataformas Windows y Linux, como en los routers que serán utilizados para realizar la interconexión entre las redes.

Finalmente se procedió a la implementación práctica del tunneling entre redes IPv6, y el tunneling hacia una red pública IPv6, con que quedó demostrado que era factible cumplir con los objetivos trazados con el presente proyecto de tesis, y que el trabajo realizado cumple con las expectativas y se ajusta a las necesidades actuales de la empresa “netXperts Consulting S.A.”.

CAPITULO I

1 INTRODUCCIÓN

1.1 Antecedentes

A pesar del éxito que ha tenido el protocolo IPv4 desde su creación en el año 1981, sus limitaciones han restringido el crecimiento de las redes que fundamentan su funcionamiento en el uso de dicho protocolo. Cuando se dieron a conocer las especificaciones de IPv4 el Internet era una comunidad de aproximadamente 100 sistemas, pero el crecimiento de la gran red mundial ha llegado a un punto en el que las especificaciones de IPv4 se están volviendo insuficientes para atender las necesidades que conllevan los avances de la tecnología.

El protocolo IPv6 fue propuesto en 1995 por el IETF[1] para combatir los inconvenientes que se han suscitado alrededor de IPv4. Las principales características de IPv6 incluyen un mayor espacio de direccionamiento, mejores prestaciones de seguridad, calidad de servicio (QoS[2]), un enrutamiento más eficiente y un mejor soporte para movilidad.

La transición entre protocolos no es nada fácil, y la transición de IPv4 a IPv6 no es la excepción. Los diseñadores de IPv6 reconocieron que dicha transición tardará varios años. Durante este período, ambos protocolos deberán

[1] IETF: Internet Engineering Task Force
[2] QoS: Quality of Service

trabajar sobre la misma infraestructura, y es por eso que se ha pensado en mecanismos que permitan viabilizar dicha convivencia. A efectos de permitir la coexistencia de ambos protocolos se idearon varias alternativas de solución, entre las cuales se destacan las siguientes:

- Asumir que todas las redes dispongan de una infraestructura totalmente compatible con IPv6, algo no muy probable debido a la alta inversión económica que esto implica.
- Disponer de una infraestructura de comunicaciones con soporte para ambos protocolos, lo que disminuiría en alto grado los costos previstos en el caso anterior.
- Manejar dos redes totalmente independientes y sin ninguna interconexión, lo que conllevaría a excesivos costos de mantenimiento de una red antigua y de implementación de una red nueva.

1.2 Justificación e Importancia

Para fines del presente proyecto se ha elegido de entre todos los mecanismos planteados para la interconexión entre los protocolos IPv4 e IPv6, a la implementación de tunneling entre redes.

El proceso de tunneling se puede explicar de la siguiente manera: cuando un paquete IPv6 está dejando un dominio IPv6 y entra en un dominio IPv4, éste se encapsula en un paquete IPv4 y se transmite a través de la red. Cuando el paquete alcanza el otro extremo de la red IPv4, el encabezado de dicho paquete es retirado y el paquete IPv6 y puede continuar hasta llegar a su destino.

La implementación de tunneling adquiere gran importancia pues permite que las redes IPv4 e IPv6 coexistan en una misma infraestructura, y puedan trabajar de forma normal, mientras se realiza el proceso de migración total hacia

IPv6. Esto implica que no habrá dificultades al momento de intercambiar información entre las mencionadas redes.

Además se puede aseverar que la implementación de túneles es una alternativa viable a corto y mediano plazo, ya que permite solucionar el problema de la comunicación entre redes utilizando los recursos existentes sin incurrir en gastos adicionales a los que deberán hacerse a futuro.

1.3 Objetivos

1.3.1 General

Implementar tunneling entre islas IPv6 a través de la red IPv4 existente, que será aplicado en la empresa “netXperts Consulting S.A.”, a fin de permitir la interconexión y la convivencia entre dichas redes.

1.3.2 Específicos

1.3.2.1 Realizar un estudio a profundidad del protocolo IPv6 para comprender su estructura, su funcionamiento y sus principales características.

1.3.2.2 Implementar tunneling desde una red IPv6 hacia otra a través de la red IPv4 existente, utilizando para ello infraestructura y equipos que se encuentren trabajando en IPv6.

1.3.2.3 Implementar tunneling desde una red IPv4 a una red IPv6 pública, por medio de la infraestructura de conectividad IPv4 que posee la empresa “netXperts Consulting S.A.”.

1.3.2.4 Efectuar las pruebas necesarias a fin de comprobar la funcionalidad del tunneling, por medio de la conexión remota a servidores IPv6.

1.4 Alcance

- Realizar un estudio a profundidad del protocolo IPv6 para comprender sus fundamentos y las principales características que lo diferencian de su antecesor; mediante una investigación en fuentes bibliográficas y sitios web con información actualizada; a fin de lograr un alto nivel de comprensión por parte de los ejecutantes del proyecto.
- Implementar tunneling desde una red IPv6 hacia otra a través de la red IPv4 existente, utilizando para ello infraestructura y equipos que se encuentren trabajando en IPv6, hasta comprobar su total funcionamiento.
- Implementar tunneling desde una red IPv4 a una red IPv6 pública, por medio de la infraestructura de conectividad IPv4 que posee la empresa "netXperts Consulting S.A.", por medio de la configuración de los equipos que trabajan con IPv6.
- Efectuar las pruebas necesarias a fin de comprobar la funcionalidad del túnel, por medio de la conexión remota a servidores IPv6 que se encuentran ya en funcionamiento; y analizar los resultados obtenidos.

1.5 Descripción de los Capítulos

Para la realización del presente proyecto de tesis se han considerado aspectos como: los fundamentos teóricos para la realización del mismo, las

configuraciones a ser realizadas tanto en hosts como en routers y la implementación del tunneling. Por esta razón se han estimado seis capítulos que se detallan a continuación:

- **Capítulo I:** Contiene toda la información previa a la realización del proyecto de tesis, incluyendo una introducción al tema así como sus antecedentes, justificación e importancia. Además se plantea el objetivo general, los objetivos específicos del proyecto, y se establece el alcance del mismo.

- **Capítulo II:** Se refiere a todos los fundamentos teóricos que se tomaron en cuenta para la realización del proyecto. Los principales temas tratados en el marco teórico tratan sobre el protocolo IPv4, las características detalladas del protocolo IPv6, y los principales mecanismos de transición para hosts y routers IPv6.

- **Capítulo III:** Se detalla las principales configuraciones a realizar tanto en hosts como en routers para permitir el soporte para el protocolo IPv6.

- **Capítulo IV:** Contiene toda la implementación del tunneling entre redes IPv6, y el tunneling hacia una red pública IPv6, así como el detalle de las configuraciones realizadas en los equipos y sus respectivas pruebas de funcionalidad.

- **Capítulo V:** Contiene las conclusiones y recomendaciones formuladas por los ejecutantes del proyecto de tesis, luego de haber finalizado con las actividades inherentes al mismo.

CAPITULO II

2 MARCO TEÓRICO

2.1 El Modelo OSI[3]

2.1.1 Antecedentes de OSI

El modelo OSI es utilizado para la visualización de entornos de red en sistemas diferentes. Los fabricantes se ajustan a dicho modelo cuando diseñan sus productos para red. El modelo OSI ofrece una descripción del funcionamiento en conjunto de hardware y software de red por niveles, para poder hacer posible las comunicaciones y desarrollar una compatibilidad total entre sistemas alrededor del mundo.

2.1.2 Estructura del Modelo OSI

- Estructura multinivel: Se diseña una estructura multinivel con la idea de que cada nivel resuelva solo una parte del problema de la comunicación, con funciones especificas.
- El nivel superior utiliza los servicios de los niveles inferiores: Cada nivel se comunica con su homologo en las otras maquinas, usando un mensaje a través de los niveles inferiores de la misma. La comunicación entre niveles se

[3] **Open System Interconnection**: Este modelo fue desarrollado por ISO (International Organization of Standarization) como una arquitectura para comunicaciones entre computadores con el objetivo de ser un protocolo estándar.

define de manera que un nivel n utilice los servicios del nivel n - 1 y proporciones servicios al nivel n + 1.

- Puntos de acceso: Entre los diferentes niveles existen interfaces llamadas “puntos de acceso” a los servicios.
- Dependencia de niveles: Cada nivel es dependiente del nivel inferior como así también lo es del nivel superior
- Encabezados: En cada nivel, se incorpora al mensaje un formato de control. Este elemento de control permite que un nivel en la computadora receptora se entere de que la emisora le esta enviando un mensaje con información.
- Número de capas: La estructura del modelo OSI define un total de 7 capas: Física, Enlace de Datos, Red, Transporte, Sesión, Presentación y Aplicación.

2.1.3 Arquitectura de red basada en el Modelo OSI

La arquitectura del modelo de referencia OSI divide el proceso de transmisión de información en siete capas, donde cada una se encarga de ejecutar una determinada parte del proceso global.

Cada nivel proporciona algún servicio o acción que prepara los datos para entregarlos a través de la red a otro equipo. Los niveles inferiores (1 y 2) definen el medio físico de la red y las tareas relacionadas, como la colocación de los bits de datos sobre las placas de red (NIC, *Network Interface Cards*) y el cable. Los niveles superiores definen la forma en que las aplicaciones acceden a los servicios de comunicación. Cuanto más alto es el nivel, más compleja es su tarea.

La interacción entre niveles adyacentes ocurre a través de una interfaz, la misma que define los servicios ofrecidos por el nivel inferior para el nivel superior, además define cómo se accede a dichos servicios. Cada nivel se basa en los estándares y actividades del nivel inferior.

Antes de pasar los datos de un nivel a otro, éstos son divididos en paquetes o unidades de información, que se transmiten como un todo desde un dispositivo a otro sobre una red. En cada nivel, el software agrega información de formato o direccionamiento al paquete que es necesario para la correcta transmisión a través de la red.

En el extremo receptor, el paquete pasa a través de los niveles en orden inverso. Cada nivel lee y quita su cabecera correspondiente y pasa el paquete hacia el siguiente nivel superior. Cuando éste alcanza el nivel de aplicación, se encuentra en su formato original, con lo que es legible por el receptor.

Con la excepción del nivel físico, ningún nivel puede pasar información directamente a su equivalente del otro equipo, es decir, que la información del equipo emisor debe ir descendiendo por todos los niveles hasta alcanzar el nivel físico. En ese momento, la información se desplaza a través del cable de red hacia el equipo receptor y asciende por sus niveles hasta que alcanza el nivel correspondiente.

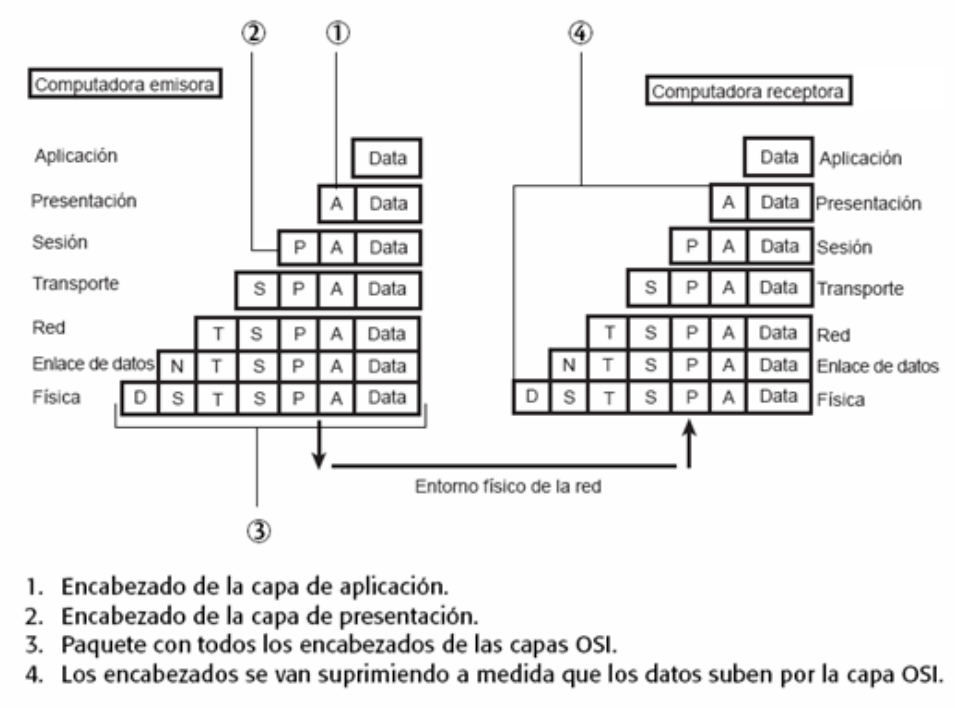


Figura 2.1: Arquitectura del modelo OSI

2.1.4 Las capas del modelo OSI

Figura 2.2: Capas del modelo OSI

2.1.4.1 La capa de aplicación

Esta capa suministra las herramientas que el usuario puede observar. También ofrece los servicios de red relacionados con estas aplicaciones de usuario, como la gestión de mensajes, la transferencia de archivos y las consultas a bases de datos. La capa de aplicación suministra cada uno de estos servicios a los distintos programas de aplicación con los que cuenta el usuario en su computadora.

2.1.4.2 La capa de presentación

La capa de presentación puede considerarse el traductor del modelo OSI. Esta capa toma los paquetes de la capa de aplicación y los convierte a un formato genérico que pueden leer todas las computadoras.

También se encarga de cifrar los datos así como de comprimirlos para reducir su tamaño. El paquete que crea la capa de presentación contiene los datos prácticamente con el formato con el que viajarán por las restantes capas de la pila OSI.

2.1.4.3 La capa de sesión

La capa de sesión establece, administra y finaliza las sesiones entre distintos hosts que se están comunicando, para realizarlo cumple tres fases:

- Establecimiento de la conexión: Los host establecen contacto, negocian las reglas de la comunicación incluyendo para ello los protocolos utilizados y los parámetros de comunicación.

- Transferencia de datos: Los hosts inician un diálogo para intercambiar datos.
- Liberación de la conexión: Cuando los hosts no van a seguir comunicados, inician la liberación ordenada de la sesión.

2.1.4.4 La capa de transporte

La capa de transporte segmenta los datos originados en el host emisor y los reensambla en el host receptor para recuperar el mensaje original.

Cuando un mensaje se divide en varios fragmentos, aumenta la posibilidad de que los mismos no se reciban en el orden correcto. Para esto la capa de transporte incluye un número de secuencia en la cabecera del mensaje que permitirá a la capa de transporte del host receptor recomponer el mensaje en el orden correcto.

Al proporcionar un servicio de comunicaciones, la capa de transporte establece, mantiene y termina adecuadamente los circuitos virtuales. Al proporcionar un servicio confiable, se utilizan dispositivos de detección y recuperación de errores de transporte.

2.1.4.5 La capa de red

La capa de red proporciona conectividad, selección de ruta, conmutación, direccionamiento y enrutamiento entre dos hosts que pueden estar ubicados en

redes geográficamente distintas. Además la capa de red se ocupa de evitar la congestión por exceso de paquetes en alguna rama de la subred.

Si el nivel de red recibe una trama que no es para la máquina en que reside, no la pasará a los niveles superiores y la reenviará hacia la máquina destino. Las rutas pueden basarse en tablas estáticas o cada paquete puede encaminarse dinámicamente en forma diferente.

2.1.4.5.1 La capa de enlace de datos

La capa de enlace de datos proporciona datos confiables que serán enviados a través del enlace físico y debe realizar dos funciones:

- Proporcionar un mecanismo de direcciones que permita entregar los mensajes en los hosts correctos, y;
- Traducir los mensajes de las capas superiores en bits que puedan ser transmitidos por la capa física.

Cuando la capa de enlace de datos recibe un mensaje, lo convierte en una trama de datos.

2.1.4.6 La capa física

La capa física se ocupa de la transmisión de bits a lo largo de un canal de comunicación, garantizando que éstos lleguen con el mismo valor al host destino. Además define las especificaciones eléctricas y mecánicas para activar, mantener

y desactivar el enlace físico entre sistemas finales. Las primeras especifican los niveles de señales para el envío de los bits y las segundas especifican conexiones físicas entre equipos, indicando la configuración de los conectores.

2.1.5 Las subcapas de la capa de enlace de datos.

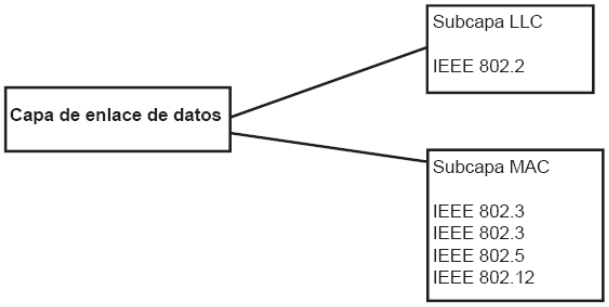


Figura 2.3: Subcapas de la capa de enlace de datos

La especificación IEEE 802 divide la capa de enlace de datos en dos subcapas: Control de Enlace Lógico (LLC[4]) y Control de Acceso al Medio (MAC[5]).

- Control de Enlace Lógico (LLC): Establece y finaliza los enlaces entre el host origen y destino cuando los datos se desplazan por el entorno físico de la red. Proporciona Puntos de Acceso al Servicio (SAP[6]), que son usados por otros equipos como referencia para poder transmitir información desde el subnivel LLC hacia los niveles superiores del modelo OSI. Además controla el tráfico de tramas, las secuencia y confirma la recepción de las mismas.
- Control de Acceso al Medio (MAC): Esta subcapa se comunica directamente con la tarjeta de red y es la responsable del envío de datos libre de errores entre dos equipos de la red. Gestiona el acceso al medio. También delimita las tramas, comprueba sus errores y reconoce sus direcciones.

[4] LLC: Logical Link Control
[5] MAC: Medium Access Control
[6] SAP: Service Access Point

2.2 El modelo TCP/IP[7]

TCP/IP es el nombre que generalmente se le da al conjunto de protocolos que se utilizan para la comunicación a través de Internet. Fue el primer conjunto de protocolos desarrollados para ser usados en este medio.

2.2.1 Arquitectura de red basada en el modelo TCP/IP

Figura 2.4: Capas del modelo TCP/IP

La arquitectura basada en TCP/IP propone cuatro capas en las que las funciones de Sesión y Presentación son responsabilidad de la capa de Aplicación y las capas de Enlace de Datos y Física son vistas como la capa de Interface de Red. Por esta razón para TCP/IP sólo existen cuatro capas, las cuales tienen un comportamiento similar a sus equivalentes en el modelo OSI tratado anteriormente.

2.2.2 Características

[7] TCP/IP: Transmission Control Protocol / Internet Protocol

- Las capas que se ven más afectadas por TCP/IP son la capa de aplicación, la capa de transporte y la capa de red. Dentro de estas capas se incluyen otros tipos de protocolos que tienen varios propósitos o funciones, todos ellos relacionados con la transferencia de información.
- TCP/IP permite la comunicación entre cualquier conjunto de redes interconectadas y sirve tanto para las comunicaciones de LAN[8] como de WAN[9]. TCP/IP incluye no sólo las especificaciones de las capas 3 y 4 sino también especificaciones para aplicaciones tan comunes como el correo electrónico, la conexión remota, la emulación de terminales y la transferencia de archivos.

2.2.3 Funcionamiento de TCP/IP

IP mueve los paquetes de datos, mientras TCP se encarga del flujo y asegura que los datos estén correctos.

Las líneas de comunicación se pueden compartir entre varios usuarios. Cualquier tipo de paquete puede transmitirse al mismo tiempo, y se ordenará y combinará cuando llegue a su destino.

Los datos no tienen que enviarse directamente entre dos computadoras. Cada paquete pasa de un dispositivo a otro hasta llegar a su destino. Los paquetes no necesitan seguir la misma trayectoria. La red puede llevar cada paquete de un lugar a otro y usar la conexión más idónea que esté disponible en ese instante. No todos los paquetes de los mensajes tienen que viajar,

[8] LAN: Local Area Network
[9] WAN: Wide Area Network

necesariamente, por la misma ruta, ni necesariamente tienen que llegar todos al mismo tiempo.

La flexibilidad del sistema lo hace muy confiable. Si un enlace se pierde, el sistema usa otro. Cuando se envía un mensaje, TCP divide los datos en paquetes, ordena éstos en secuencia, agrega cierta información para control de errores y después los lanza hacia fuera, y los distribuye. En el otro extremo, TCP recibe los paquetes, verifica si hay errores y los vuelve a combinar para convertirlos en los datos originales. De haber error en algún punto, TCP envía un mensaje solicitando que se vuelvan a enviar determinados paquetes.

2.3 Comparación entre la pila de protocolos OSI y TCP/IP

Aunque el modelo OSI es universalmente reconocido y se pensaba que iba a ser seguido por la mayoría de las redes futuras el estándar abierto de Internet ha hecho que la pila de protocolos basada en TCP/IP sea la más utilizada en la actualidad ya que hace posible la comunicación entre computadoras desde cualquier parte del mundo.

El modelo OSI se definió antes que los protocolos, mientras que en TCP/IP se definieron primero los protocolos y el modelo fue en realidad una descripción de los protocolos existentes.

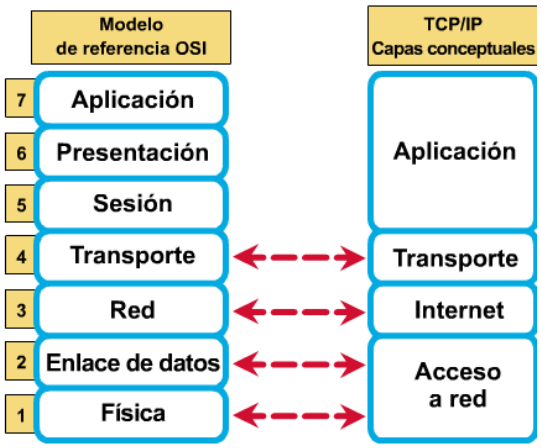


Figura 2.5: Comparación entre los modelos OSI y TCP/IP

El modelo TCP/IP destaca una mayor flexibilidad en la capa de aplicación; en la capa de transporte involucra dos protocolos: TCP y UDP[10]; en la capa de Internet involucra un solo protocolo: IP; y la capa inferior, está relacionada con la tecnología LAN o WAN que se utilizará.

2.4 El Protocolo IPv4

2.4.1 Introducción

El protocolo IPv4 está diseñado para su uso en sistemas interconectados de redes de comunicación de ordenadores por intercambio de paquetes. El protocolo IPv4 proporciona los medios necesarios para la transmisión de bloques de datos llamados datagramas desde el origen al destino. El protocolo IPv4

[10] UDP: User Datagram Protocol

también se encarga, si es necesario, de la fragmentación y el reensamblaje de grandes datagramas para su transmisión a través de redes de trama pequeña.

2.4.2 Operación

El protocolo IPv4 implementa dos funciones básicas: direccionamiento y fragmentación. Los módulos IPv4 usan las direcciones que se encuentran en la cabecera para transmitir los datagramas hacia sus destinos. La selección de un camino para la transmisión se llama encaminamiento. El protocolo IPv4 usa cuatro mecanismos clave para prestar su servicio:

- Tipo de Servicio: Se utiliza para indicar la calidad del servicio deseado. El tipo de servicio es un conjunto abstracto o generalizado de parámetros que caracterizan las elecciones de servicio presentes en las redes que forman Internet
- Tiempo de Vida: Es una indicación de un límite superior en el periodo de vida de un datagrama. Es fijado por el remitente del datagrama y reducido en los puntos a lo largo de la ruta donde es procesado. Si el tiempo de vida se reduce a cero antes de que el datagrama llegue a su destino, el datagrama es destruido, algo que no sucede en una red que se encuentre operando correctamente y fue diseñado para evitar que un paquete huérfano esté presente en la red de manera indefinida.
- Opciones: Proporcionan funciones de control necesarias que incluyen recursos para marcas de tiempo, seguridad y encaminamiento especial.

- Suma de Control de Cabecera: Proporciona una verificación de que la información utilizada al procesar el datagrama ha sido transmitida correctamente. Si la suma de control de cabecera falla, el datagrama es descartado inmediatamente por la entidad que detecta el error.

El protocolo IPv4 no proporciona ningún mecanismo de comunicación fiable. No existen acuses de recibo (ni entre extremos ni entre saltos), no hay control de errores para los datos, sólo una suma de control de cabecera, no hay retransmisiones y tampoco existe control de flujo. Los errores detectados pueden ser notificados por medio del Protocolo de Mensajes de Control de Internet (ICMP[11]).

2.4.3 Descripción de Funciones

La principal función del protocolo IPv4 es mover datagramas a través de un conjunto de redes interconectadas. Esto se consigue pasando los datagramas desde un módulo IPv4 a otro hasta que se alcanza el destino. Los módulos IPv4 residen en hosts y gateways en el sistema Internet. Los datagramas son encaminados desde un módulo IPv4 a otro a través de redes individuales basándose en la interpretación de una dirección IPv4.

- Direccionamiento: Primeramente debe establecerse una diferencia entre los términos: nombre, dirección y ruta. Un nombre indica qué es lo que se busca, una dirección indica dónde está y una ruta indica cómo llegar hasta dicho lugar. El protocolo IPv4 maneja principalmente direcciones; es tarea de los

[11] ICMP: Internet Control Message Protocol

protocolos de menor nivel realizar la correspondencia entre direcciones de red y rutas.

- Fragmentación: La fragmentación de un datagrama es necesaria cuando éste se origina en una red que permite un tamaño de paquete grande y debe atravesar una red que limita los paquetes a un tamaño inferior para llegar a su destino. Un datagrama puede ser marcado mediante una bandera para indicar a los involucrados en la comunicación de datos que se trata de un datagrama que no necesita ser fragmentado. Si un datagrama de estos no puede ser entregado en su destino sin fragmentarlo, entonces debe ser descartado.

2.4.4 Especificación

2.4.4.1 Formato de la Cabecera IPv4

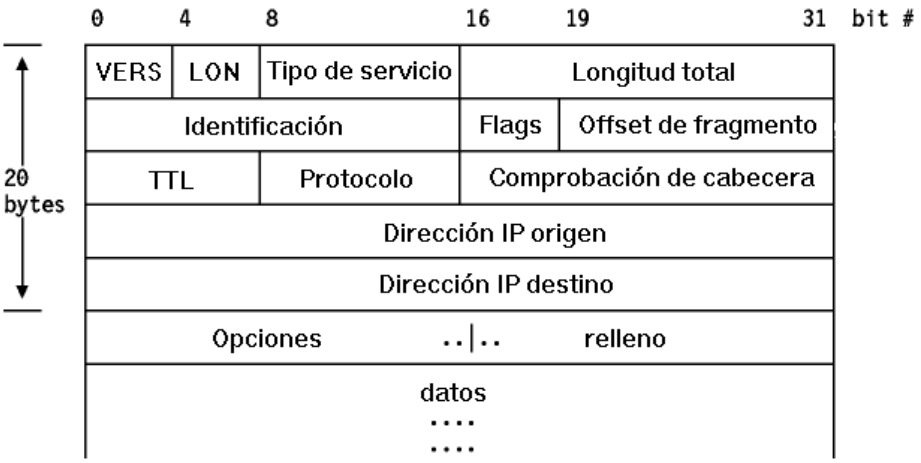


Figura 2.6: Formato de la cabecera IPv4

- VERS → 4 bits. Versión. Permite que en una misma red puedan coexistir paquetes de distintas versiones del protocolo IP.

- LON → 4 bits. Longitud. También conocido como IHL[12]. Especifica la longitud de la cabecera, la misma que puede variar debido a la presencia de campos opcionales. La longitud de la cabecera siempre ha de ser un número entero de palabras de 32 bits, por lo que si la longitud de los campos opcionales no es un múltiplo exacto de 32 bits se añade un relleno al final de la cabecera.
- Tipo de Servicio → 8 bits. Proporciona una indicación de los parámetros de la calidad de servicio deseada. Algunas redes ofrecen prioridad de servicio, la cual trata de algún modo el tráfico de alta prioridad como más importante que el resto del tráfico. La elección más común es un compromiso a tres niveles entre baja demora, alta fiabilidad, y alto rendimiento.

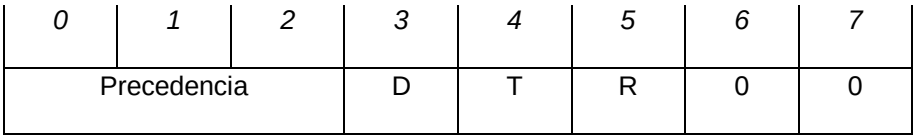


Figura 2.7: Descripción del campo Tipo de Servicio

Tabla 2.1: Descripción del Tipo de Servicio

Bits 0 – 2	:	Precedencia	
Bit 3	:	0 = Demora Normal	1 = Baja Demora
Bit 4	:	0 = Rendimiento Normal	1 = Alto Rendimiento
Bit 5	:	0 = Fiabilidad Normal	1 = Alta Fiabilidad
Bits 6 – 7	:	Reservado para uso futuro	

- Longitud Total → 16 bits. Es la longitud del datagrama, medida en octetos, incluyendo la cabecera y los datos. Todos los hosts deben estar preparados para aceptar datagramas de hasta 576 octetos y es recomendable que los

[12] IHL: Internet Header Lenght – Longitud de Cabecera Internet

hosts envíen datagramas de mayor tamaño sólo si se tiene la seguridad de que el destinatario está preparado para aceptarlos.

- Identificación → 6 bits. Es un valor asignado por el remitente a cada uno de los fragmentos (si los hubiese) como ayuda para el proceso de reensamblaje del datagrama.
- Flags (indicadores) → 3 bits. Son diversos indicadores de control.

0	*1*	*2*
0	DF	MF

Figura 2.8: Descripción del campo Flags

Tabla 2.2: Descripción de los Indicadores

Bit 0	:	Reservado (debe ser cero)	
Bit 1	:	(DF) Don't Fragment (No Fragmentar)	
		0 = Puede fragmentarse	1 = No fragmentar
Bit 2	:	(MF) More Fragments (Más Fragmentos)	
		0 = Último fragmento	1 = Más fragmentos

- Offset de Fragmento → 13 bits. Posición de Fragmento. Indica a que parte del datagrama pertenece un fragmento. La posición se mide en unidades de 8 octetos. El primer fragmento tiene la posición 0.
- TTL[13] → 8 bits. Tiempo de Vida. Sirve para descartar un datagrama cuando ha dado un número excesivo de saltos o ha pasado un tiempo excesivo viajando por la red y es presumiblemente inútil. Cada router por el que pasa dicho datagrama está obligado a restar una unidad del valor del TTL, independientemente del tiempo que tarde en reenviarlo. Esto evita que por

[13] TTL: Time To Live

algún problema en las rutas se produzcan bucles y un datagrama permanezca indefinidamente en la red.

- Protocolo → 8 bits. Especifica a que protocolo del nivel de transporte corresponde el datagrama. La siguiente tabla muestra algunos de los posibles valores de este campo.

Tabla 2.3: Ejemplo de valores y significados del campo Protocolo

Valor	Protocolo	Descripción
0		Reservado
1	ICMP	Internet Control Message Protocol
2	IGMP	Internet Group Management Protocol
3	GGP	Gateway-to-Gateway Protocol
4	IP	IP en IP (encapsulado)
5	ST	Stream
6	TCP	Transmission Control Protocol
8	EGP	Exterior Gateway Protocol
17	UDP	User Datagram Protocol
29	ISO-TP4	ISO Transport Protocol Clase 4
38	IDRP-CMTP	IDRP Control Message Transport Protocol
80	ISO-IP	ISO Internet Protocol (CLNP)
88	IGRP	Internet Gateway Routing Protocol (Cisco)
89	OSPF	Open Shortest Path First
255		Reservado

- Comprobación de Cabecera → 16 bits. También denominado Checksum, sirve para detectar errores producidos en la cabecera del datagrama. El checksum solo cubre la cabecera del datagrama, más no los datos. El campo checksum se ha de recalcular en cada salto, ya que al menos el TTL cambia.
- Dirección IP Origen → 32 bits. Valor de la dirección IP de origen
- Dirección IP Destino → 32 bits. Valor de la dirección IP de destino
- Opciones → variables. Los campos opcionales de la cabecera no siempre están soportados en los routers y no se utilizan con frecuencia. Cada campo opcional está compuesto por una etiqueta, seguida opcionalmente de información adicional. El uso de los campos opcionales tiene generalmente problemas de rendimiento, razón por la cual solo se los utiliza en situaciones de prueba o diagnóstico de errores, más no en entornos en producción.
- Valor de Relleno → variable. El valor de relleno se usa para asegurar que la cabecera IP ocupe un múltiplo de 32 bits. El valor del relleno es cero.

2.4.5 Direcciones IPv4

Cada interfaz de red de cada nodo de una red se identifica mediante al menos una dirección única de 32 bits. Las direcciones IP se suelen representar por cuatro números decimales separados por puntos, que equivalen al valor de cada uno de los cuatro bytes que componen la dirección. Por ejemplo una dirección IPv4 válida sería 147.156.23.208. Si un nodo dispone de varias interfaces físicas cada una de ellas deberá tener necesariamente una dirección IPv4 distinta si se desea que sea accesible de forma diferenciada para este protocolo.

Las direcciones IPv4 tienen una estructura jerárquica, en la cual una parte de la dirección corresponde a la red, y la otra al host dentro de la red. Cuando un router recibe un datagrama por una de sus interfaces compara la parte de red de la dirección con las entradas contenidas en sus tablas y envía el datagrama por la interfaz correspondiente. En el diseño inicial de Internet se reservaron los ocho primeros bits para la red, dejando los 24 restantes para el host; pero luego debido al amplio crecimiento de la red mundial se reorganizó el espacio de direcciones. El resultado de dicha reorganización es lo que hoy se conoce como clases de redes ó clases de direcciones.

2.4.6 Clases de Direcciones

En IPv4 se han definido cinco clases de direcciones para ordenar redes de varios tamaños. Básicamente se manejan tres clases de direcciones: A, B y C. La clase de una dirección define cuántos bits se usan para el identificador de red (Red ID) y cuántos bits se usan para el identificador de host (Host ID). La clase también define el posible número de redes y el número de hosts por red. La identificación de una clase de dirección se la puede hacer observando el número del primer octeto.

Tabla 2.4: Identificadores de Red y Host de las clases A, B y C

Clase	Dirección IP	Red ID	Host ID
A	w.x.y.z	w	x.y.z
B	w.x.y.z	w.x	x.y

C	w.x.y.z	w.x.y	z

Existen otras dos clases denominadas D y E. Las direcciones de clase D son destinadas para grupos multicast. Un grupo multicast puede contener uno o más hosts, ó ninguno en absoluto. Los cuatro bits de mayor orden en las direcciones de clase D tiene siempre el valor 1110, mientras que los bits restantes designan el grupo en el cual el cliente participa. Las direcciones de clase E son experimentales y por lo tanto no están disponibles para su uso general. Los cuatro bits de mayor orden en las direcciones de clase E tienen en valor 1111.

2.4.7 Consideraciones sobre Direccionamiento

A pesar de que no existen reglas sobre cómo asignar direcciones IP es necesario conocer cierto tipo de consideraciones para evitar posteriores inconvenientes con los identificadores de red y hosts. Entre las principales consideraciones tenemos:

- El identificador de red no puede ser 127. Este ID está reservado para funciones de loopback y diagnóstico.
- Los bits del identificador de red y de host no pueden ser todos 1. Si todos los bits tienen el valor de 1 la dirección será interpretada como un broadcast en lugar de un host ID.
- Los bits del identificador de red y de host no pueden ser todos 0. Si todos los bits tienen el valor de 0 la dirección será interpretada como un identificador de la red actual.

2.4.8 Asignación de Identificadores de Red

Se requiere de un único identificador para cada red y conexión de área extendida. Si se planea tener una conexión a redes públicas, como por ejemplo el Internet es necesario obtener un identificador de red válido, en este caso por parte del InterNIC[14]. Si no se considera la conexión a una red pública se puede utilizar cualquier identificador de red válido, pero para evitar un caos en la asignación de redes se han reservado bloques de direcciones privadas las que no puede ser ruteadas, estas son la 10, 176 y 192.

Un identificador de red determina los hosts que están situados en una misma red física. A todos los hosts de la misma red física se les debe asignar el mismo identificador de red a fin de poder establecer comunicación entre los mismos. Si las redes están conectadas por medio de routers, se requiere un único identificador de red por cada conexión WAN. Por ejemplo en la figura 2.9:

- Las redes 1 y 3 representan dos redes ruteables
- La red 2 representa la conexión WAN entre los routers
- La red 2 requiere un identificador de red para que las interfases entre los dos routers puedan tener identificadores de host únicos.

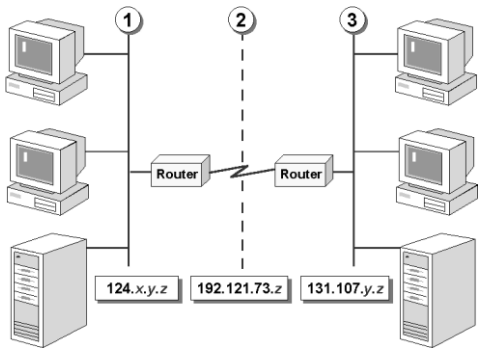


Figura 2.9: Identificadores de red para la conexión entre distintas redes

[14] InterNIC: Internet Network Information Center

2.4.9 Asignación de Identificadores de Host

Un identificador de host distingue un equipo dentro de una red y debe ser único para toda la misma. Todos los hosts TCP/IP, incluyendo las interfases hacia los routers, requieren de identificadores únicos. El identificador del router es la dirección IP configurada como puerta de salida (gateway) predeterminada para las estaciones de trabajo. En el ejemplo anterior, para el host de la red 1 con la dirección 124.0.0.27, el gateway predeterminado es 124.0.0.1.

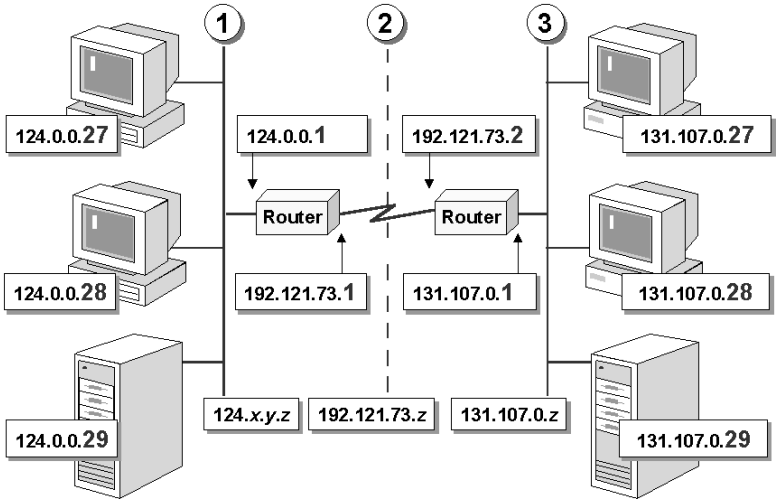


Figura 2.10: Identificadores de Host para las distintas redes

2.4.10 Subredes

La visión original del Internet consideraba una jerarquía de dos niveles: en el nivel superior el Internet como un todo, y en el nivel inferior redes individuales, cada una con su propio número de red. A pesar de que este punto de vista probó su simplicidad y poder, hubo una cantidad de organizaciones que lo encontraron

inadecuado y creyeron conveniente agregar un tercer nivel a la interpretación de las direcciones de Internet. Bajo este criterio una red puede ser dividida en un conjunto de subredes.

El modelo de tres niveles es útil en redes pertenecientes a organizaciones relativamente grandes donde a menudo es necesario usar más de una red LAN para cubrir un área local. Cada LAN debe ser tratada como una subred.

El procedimiento de subredes, también conocido como subnetting, divide el campo de host en campos de subredes y hosts, creando una dirección de tres partes. El campo de red permanece sin cambios y es determinado por medio del direccionamiento de clases. El límite entre los campos de subred y host es determinado usando una submáscara de red. Los bits de la submáscara de red tienen el valor 1 cuando identifican bits en los campos de red y subred, y el valor 0 cuando identifican bits en el campo de hosts.

Con el tiempo el subnetting se convirtió en la alternativa más utilizada para el manejo del espacio de direcciones de red. La principal razón es la flexibilidad que permite a cualquier red ser subdividida en partes más pequeñas que son mucho más administrables y funcionales, de acuerdo a las necesidades y requerimientos de las organizaciones.

2.4.11 Protocolos de Control de Internet

Normalmente los datagramas, que son los que transportan todo el tráfico de Internet, transportan unidades de datos de los protocolos de transporte utilizados en Internet. Todas las aplicaciones de Internet generan tráfico que comúnmente es TCP ó UDP. Sin embargo, en el campo protocolo de la cabecera del datagrama existen otros posibles valores. Algunos de los datos que pueden transportarse en datagramas IP son mensajes de protocolos de control de Internet, los cuales juegan un papel importante en el correcto funcionamiento de la red. El protocolo ICMP es el más conocido y utilizado de los protocolos de control de Internet.

2.4.11.1 ICMP (Internet Control Message Protocol)

En el funcionamiento normal de una red se dan a veces situaciones extraordinarias que requieren enviar avisos especiales. El mecanismo para reportar todos estos incidentes en Internet es el protocolo conocido como ICMP.

Conviene recordar que los mensajes ICMP viajan por la red como datagramas IP. Estos mensajes son generados por el host o router que detecta el problema o situación extraordinaria y se dirigen hacia el host o router que aparece en el campo dirección origen del datagrama que causó el problema. Para facilitar la identificación del datagrama por parte del host emisor la mayoría de los mensajes ICMP incluyen, además del código de error correspondiente, la cabecera y los primeros ocho bytes de datos del datagrama original.

A continuación se describen los mensajes ICMP más importantes:

- Destination Unreachable: Se produce cuando no se puede entregar el datagrama a su destino por diversas situaciones que pueden ocurrir en la red ó en el medio de transmisión.
- Source Quench: Se creó para permitir a los routers solicitar una reducción en el tráfico generado por los hosts en caso de congestión, aunque en la práctica su uso agrava los problemas de congestión.
- Echo Request y Echo Reply: Se usan para detectar si un destino determinado está operativo. El emisor genera un mensaje del tipo Echo Request (pedido), y al recibir el mensaje el destinatario debe responder con el comando Echo Reply (respuesta).
- Time Exceded: Se envía al emisor cuando un paquete es descartado porque su TTL ha llegado a cero. Esto puede ser síntoma de que se ha producido algún bucle en la red, o que el valor del TTL utilizado es demasiado bajo para el diámetro de la red.
- Redirect: Se utiliza para alertar al host emisor cuando se sospecha que un paquete se está encaminando incorrectamente.

2.4.12 Protocolos de Routing

La red Internet está formada por muchas redes interconectadas, pertenecientes a todo tipo de empresas y organizaciones. Todas estas redes interconectadas comparten a nivel de red el protocolo IP. Al margen de esta

interoperabilidad existen dos aspectos fundamentales en los que las redes pueden diferir: el protocolo de routing utilizado, y la política de intercambio de tráfico.

2.4.13 Protocolos de Routing Interno

Los protocolos de routing interno pueden agruparse en protocolos de vector distancia, entre los que se destacan RIP, IGRP y EIGRP; y protocolos de estado del enlace, donde los más importantes son IS-IS y OSPF.

2.4.13.1 RIP[15] y RIPv2

RIP es uno de los protocolos de routing más antiguos y sufre los problemas típicos de los algoritmos basados en el vector distancia, tales como la cuenta a infinito. Algunos de estos problemas aumentan a medida que crece el tamaño de los sistemas autónomos, por lo que en la práctica no es aconsejable usar RIP en ninguna red que tenga más de 5 a 10 routers.

2.4.14 Protocolos de Routing Externo

En años anteriores se utilizaba como protocolo de routing externo el EGP[16], diseñado entre 1982 y 1984. Como era de esperar EGP no fue capaz de soportar la enorme evolución que sufrió Internet y por ende el IETF desarrolló un nuevo

[15] RIP: Routing Information Protocol
[16] EGP: Exterior Gateway Protocol

protocolo de routing externo, denominado BGP[17]. La primera especificación de BGP apareció en 1989; desde entonces el IETF ha producido cuatro versiones de dicho protocolo.

Los routers que utilizan BGP forman entre ellos una red e intercambian información de routing para calcular las rutas óptimas; se utiliza el vector distancia, pero para evitar el problema de la cuenta a infinito la información intercambiada incluye, además de los routers accesibles y el costo, la ruta completa utilizada para llegar a cada posible destino; de esta forma el router que recibe la información puede descartar las rutas que pasan por él mismo que son las que podrían dar lugar al problema de la cuenta a infinito. La especificación de la ruta completa permite también a los routers revisar si dicha ruta es conforme con las políticas que se hayan especificado en cuanto a tránsito por otros sistemas autónomos. BGP permite introducir manualmente restricciones o políticas; éstas se traducen en que cualquier ruta que viola la regla recibe automáticamente una distancia de infinito.

2.5 El Protocolo IPv6

2.5.1 Introducción

El protocolo IPv6 incorpora nuevas características a la probada funcionalidad de su predecesor, cubriendo de esa forma las necesidades de los

[17] BGP: Border Gateway Protocol

usuarios y aplicaciones actuales, tales como un mayor espacio de direccionamiento, seguridad, movilidad y una mejor calidad de servicio.

A continuación se detallan las principales características que ofrece el protocolo IPv6:

- Aumento del espacio de direcciones: El protocolo IPv4 está basado en una arquitectura que utiliza direcciones de 32 bits (4 octetos). Con la nueva versión del protocolo, las direcciones constan de 128 bits, con lo que se soluciona el problema del agotamiento de direcciones IPv4.
- Autoconfiguración: En el momento que un host se conecta a una red recibe los datos necesarios para empezar a comunicarse. Los routers proveen de información a todos los nodos sobre un enlace local, por lo tanto un host puede autoconfigurarse a sí mismo con la información proporcionada y con su dirección MAC.
- Movilidad: La movilidad está llegando a ser una característica importante y crítica en las redes actuales. Es un estándar que permite a los dispositivos móviles desplazarse sin perder las conexiones existentes.
- Seguridad: Mientras el uso de IPSec[18] es opcional en IPv4, el mismo es una característica incorporada en IPv6. Por eso, los diseñadores de las redes podrían habilitar IPSec en todos los nodos IPv6, haciendo de esta manera más segura a las redes.
- Encaminamiento jerárquico: El encaminamiento bajo IPv6 es jerárquico y sin clases. Con esto se pretende conseguir la disminución en el tamaño de las tablas de rutas en los backbones haciendo más simples las tareas de enrutamiento.

[18] IPSec: Internet Protocol Security

- Calidad de servicio (QoS): El protocolo IPv6 dispone de campos más amplios para definir la prioridad y flujo de cada paquete. Según el contenido de este campo, el router deberá darle un trato más o menos especial.

2.5.2 Descripción

2.5.2.1 La cabecera IPv6

La cabecera de un paquete IPv6 es más sencilla que la del paquete IPv4, con la ventaja de que presenta una mayor funcionalidad. La cabecera de un paquete IPv4 es variable, por lo que necesita un campo de tamaño o longitud (lenght). Sin embargo, para simplificar la vida de los routers, IPv6 utiliza un tamaño de cabecera fijo de 40 bytes, que componen un total de ocho campos:

- Versión (4 bits), al igual que en la cabecera IPv4 este campo identifica la versión IP del paquete de datos. Para el caso de IPv6 este campo tiene el valor 0110.
- Clase de tráfico (8 bits), para poder diferenciar entre servicios sensibles a la latencia, como por ejemplo voz sobre IP (VoIP[19]), de otros que no necesitan prioridad, como tráfico http[20].
- Etiqueta de flujo (20 bits), permite la diferenciación de flujos de tráfico. Esto tiene importancia a la hora de manejar la calidad de servicio (QoS)
- Tamaño de payload (16 bits), describe el tamaño en octetos de la sección de datos del paquete. Al ser este campo de 16 bits, podremos usar paquetes de hasta más de 64000 bytes.

[19] VoIP: Voice Over Internet Protocol
[20] Http: Hypertext Transfer Protocol

- Siguiente cabecera (8 bits), Se utiliza para indicar el protocolo al que corresponde la cabecera que se sitúa a continuación de la actual. El valor de este campo es el mismo que el del campo protocolo en la cabecera IPv4.
- Límite de saltos (8 bits), especifica el número de saltos de router que puede hacer el paquete antes de ser desechado. Con 8 bits podremos tener un máximo de 255 saltos.
- Dirección origen y de destino (128 bits cada una), son las direcciones de los nodos IPv6 que realizan la comunicación.

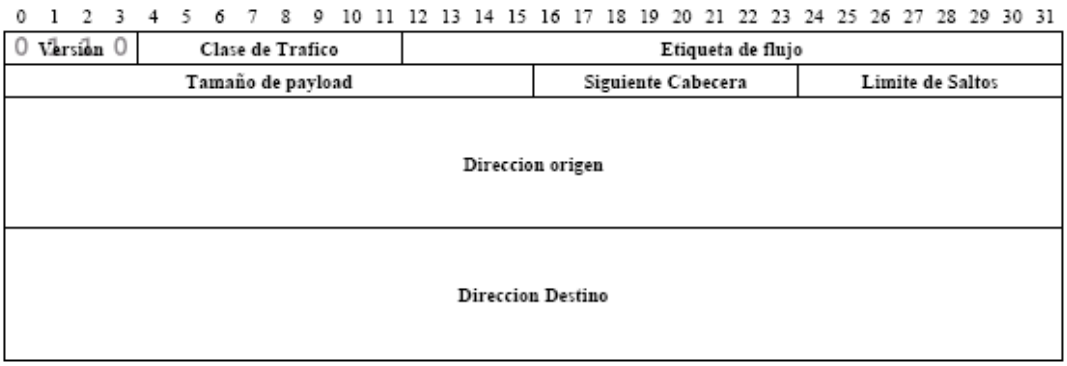


Figura 2.11: Cabecera IPv6

2.5.2.2 Campo de Siguiente Cabecera (Next Header Field)

Dentro de una cabecera de IPv6 existe un campo llamado de siguiente cabecera que permite describir con más detalle las opciones del paquete. Esto quiere decir que en realidad se tiene una cabecera de tamaño fijo por norma general y otra cabecera de tamaño variable en caso de que se utilice alguna de las características avanzadas.

En el campo de siguiente cabecera se codificarán las opciones presentes en la siguiente cabecera:

Tabla 2.5: Valores y significado del campo siguiente cabecera

Siguiente cabecera	Valor del campo
Opciones de Hop-by-Hop	0
Opciones de destino	60
Encaminamiento	45
Fragmento	44
Autenticación	51
Encapsulación	50
Ninguna	59

Esta arquitectura es muy flexible, ya que cada cabecera tiene un campo de siguiente cabecera, con lo que se puede tener varias opciones agregadas; por ejemplo:

Figura 2.12: Cabecera IPv6 básica y datos

Figura 2.13: Cabecera IPv6 básica, fragmento y datos

Con la cabecera de encaminamiento se consigue la funcionalidad equivalente de IPv4 de Source-Routing, es decir, especificar los nodos intermedios por los que ha de pasar el paquete.

Un aspecto a recalcar es que los nodos intermedios o encaminadores no deben examinar más que la cabecera IPv6 básica. Existen excepciones como en el caso de que existan cabeceras de opciones Hop-by-Hop o, como en el caso anterior, que exista una cabecera de encaminamiento en el que solo los nodos en ella definidos deberán alterar el paquete.

La especificación recomienda además el siguiente orden para las cabeceras adicionales:

- Cabecera IPv6 básica
- Opciones Hop-by-Hop
- Opciones de destino
- Encaminamiento
- Fragmento
- Autenticación
- Encapsulación
- Opciones de destino
- Cabecera nivel superior

Las opciones de destino pueden ser procesadas en momentos distintos dependiendo de si el paquete atraviesa un nodo intermedio o llega al nodo destino. La única restricción de la especificación es que las opciones de Hop-by-Hop han de ir siempre de la cabecera básica.

2.5.3 Direccionamiento IPv6

2.5.3.1 Espacio de direcciones en IPv6

La característica más obvia que distingue a IPv6 es el uso de direcciones mucho más largas. El tamaño de una dirección en IPv6 es de 128 bits, es decir, cuatro veces más larga que una dirección IPv4.

Con IPv6 es muy difícil concebir que el espacio de direcciones se agote. Para ilustrar esta aseveración basta con mirar los siguientes datos:

Número de direcciones en IPv4:	4,294,967,296
Población del planeta (2001):	6,170,000,000
Número de direcciones en IPv6:	340,282,366,920,938,463,463,374,607,431,768,211,456

Figura 2.14: Espacio de direccionamiento en IPv4 e IPv6

El tamaño relativo de una dirección IPv6 fue diseñado a fin de poder subdividirla dentro de dominios de ruteo jerárquicos que reflejan la topología de la era moderna del Internet. El uso de 128 bits permite tener múltiples niveles de jerarquía y flexibilidad en el diseño jerárquico de direccionamiento y ruteo, lo que no se tiene en el actual Internet basado en IPv4.

2.5.3.2 Sintaxis de las direcciones en IPv6

Las direcciones IPv4 están representadas en un formato decimal separado por puntos. Estas direcciones de 32 bits están divididas en límites de 8 bits. Cada conjunto de 8 bits es convertido a su equivalente decimal y separado por puntos.

En IPv6, para las direcciones de 128 bits existen divisiones en límites de 16 bits, y cada bloque de 16 bits es convertido a un número hexadecimal y separado por dos puntos (:). La representación resultante es llamada *hexadecimal-dos puntos* (colon-hexadecimal).

La siguiente es una dirección IPv6 en forma binaria:

0010000111011010000000001101001100000000000000000001011110011101100000101010101000000000011111111111111100010100010011100010110 10

La dirección de 128 bits dividida en límite de 16 bits es la siguiente:

0010000111011010 0000000011010011 0000000000000000 0010111100111011 0000001010101010 0000000011111111 1111111000101000 1001110001011010

Cada bloque de 16 bits es convertido a hexadecimal y delimitado por dos puntos. El resultado es:

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

La representación IPv6 puede ser adicionalmente simplificada removiendo los ceros más significativo (a la izquierda) existentes dentro de cada bloque de 16 bits. Sin embargo cada bloque debe tener al menos un dígito. Con la supresión de ceros, la representación de la dirección queda de la siguiente forma:

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

Algunos tipos de direcciones contienen largas secuencias de ceros. Para favorecer la simplificación en la representación de las direcciones, una secuencia

continua de bloques de 16 bits con valor 0 puede ser compactada utilizando el símbolo "::" conocido como doble dos puntos (double colon). Por ejemplo, la dirección de enlace local FE80:0:0:0:2AA:FF:FE9A:4CA2 puede ser compactada así: FE80::2AA:FF:FE9A:4CA2. La dirección multicast FF02:0:0:0:0:0:0:2 puede ser representada así: FF02::2. La compactación de ceros puede ser solamente utilizada una vez en una dirección dada.

2.5.4 Prefijos IPv6

El prefijo es la parte de la dirección que indica los bits que tienen valores estables ó son los bits del identificador de red. Los prefijos IPv6 para los identificadores de subred, rutas, y rangos de direcciones están expresados de la misma forma que la notación CIDR[21] para IPv4. Un prefijo IPv6 está escrito en la notación *dirección/longitud de prefijo*. Por ejemplo, 21DA:D3::/48 es un prefijo de ruta y 21DA:D3:0:2F3B::/64 es un prefijo de subred.

Cabe recalcar que las implementaciones IPv4 usan comúnmente una representación decimal con puntos del prefijo de red conocido como máscara de subred. En IPv6 se utiliza la longitud del prefijo para determinar jerarquías en las direcciones, con lo que el concepto de la máscara de subred no tiene la importancia que se le da en IPv4.

2.5.5 Tipos de direcciones IPv6

Existen 3 tipos de direcciones en IPv6:

[21] CIDR: Classless Inter-Domain Routing

- Unicast: Una dirección unicast identifica una interfase única dentro del ámbito del tipo de direcciones unicast. Con la topología apropiada para ruteo unicast, los paquetes direccionados a una dirección unicast son entregados a una sola interfase.
- Multicast: Una dirección multicast identifica múltiples interfases. Con la topología apropiada para ruteo multicast, los paquetes direccionados a una dirección multicast son entregados a todas las interfases que son identificadas por la dirección. Las direcciones multicast son usadas para comunicaciones uno-a-muchos, con entrega a múltiples interfases.
- Anycast: Una dirección anycast identifica múltiples interfases. Con la topología apropiada de ruteo, los paquetes direccionados a una dirección anycast son entregados a una sola interfase, la interfase más cercana que es identificada por la dirección. La interfase más cercana es definida en términos de distancia de ruteo. Las direcciones anycast son usadas para comunicaciones uno-a-uno-de-muchos, con entrega a una sola interfaz.

En todos los casos, las direcciones IPv6 identifican interfases más no a nodos. Un nodo es identificado por alguna dirección unicast asignada a una de sus interfases.

2.5.6 Direcciones Unicast

2.5.6.1 Direcciones Globales Unicast

Las direcciones globales unicast son equivalentes a las direcciones públicas de IPv4. Son globalmente ruteables y accesibles en la porción IPv6 de Internet. A diferencia del Internet actual, basado en IPv4, que es una mezcla de ruteo plano y jerárquico, el Internet basado en IPv6 ha sido diseñado desde su creación para soportar un ruteo y direccionamiento eficiente y jerárquico. El ámbito de la dirección global unicast, definido como la región de trabajo en Internet IPv6, es todo el Internet IPv6.

La siguiente figura muestra la estructura de una dirección unicast global que es actualmente asignada para la Internet IPv6.

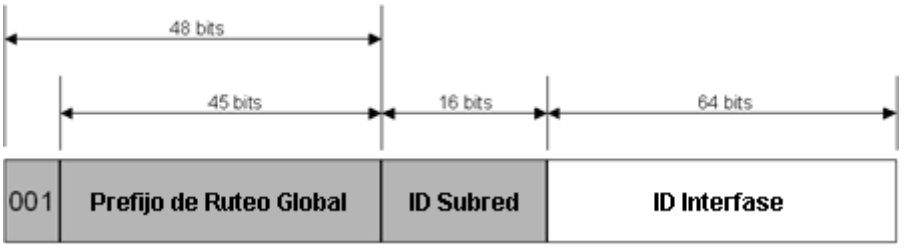


Figura 2.15: Dirección Global Unicast definida en el RFC 3587[22]

Los campos de una dirección global unicast son los siguientes:

- Porción fija: con valor 001. Los tres bits de mayor orden tienen el valor 001. Los prefijos reservados para direcciones de este tipo son: 2xxx: y 3xxx: aunque dentro de estas hay mas subdivisiones (3ffe:, 2002:, etc.) usadas para distintos propósitos.
- Prefijo de Ruteo Global: Indica el prefijo de ruteo global para el sitio específico de una organización. La combinación de 3 bits fijos y el prefijo

[22] RFC 3587: IPv6 Global Unicast Address Format

de 45 bits, es usada para crear un prefijo de 48 bits, el cual es asignado al sitio de la organización. Una vez asignado, los routers de la Internet IPv6 envían tráfico coincidente con el prefijo de 18 bits hacia los routers de la organización.

- ID Subred: El ID de Subred es usado dentro del sitio de una organización para identificar subredes. El tamaño de este campo es de 16 bits. Dicho sitio puede usar estos 16 bits para crear 65,536 subredes ó múltiples niveles de jerarquía de direccionamiento y una eficiente infraestructura de ruteo.
- ID Interfase: Indica la interfase en una subred específica dentro del sitio. El tamaño de este campo es de 64 bits. Los campos dentro de la dirección global unicast crean una estructura de tres niveles como se muestra a continuación:

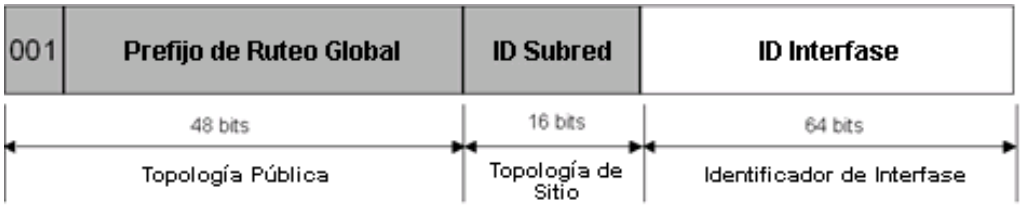


Figura 2.16: Estructura de tres niveles de una dirección global unicast

La topología pública es la colección de ISP's[23], grandes y pequeños, que proveen acceso al Internet IPv6. La topología del sitio es la colección de subredes dentro de la organización. El identificador de interfase se refiere a una interfase específica en una subred dentro de la organización.

[23] ISP: Internet Service Provider – Proveedor de Servicios de Internet

2.5.6.2 Direcciones Unicast de Uso Local

Existen dos tipos de direcciones unicast de uso local: direcciones de Enlace Local y direcciones de Sitio Local

- Direcciones de Enlace Local: Las direcciones de enlace local son usadas por nodos en la comunicación con nodos vecinos en el mismo enlace. Por ejemplo, en una red IPv6 de enlace simple sin un router, las direcciones de enlace local son usadas para comunicarse entre hosts en el enlace. El ámbito de una dirección de enlace local es el enlace local.

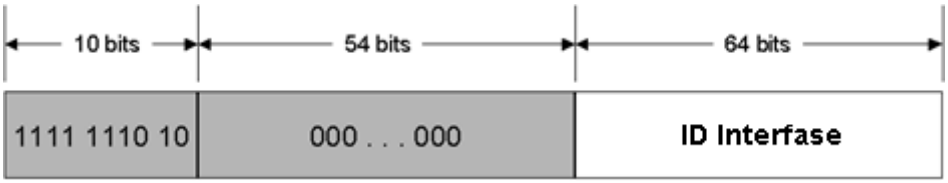


Figura 2.17: Dirección de enlace local

Las direcciones de enlace local siempre comienzan con FE80. Con el identificador de interfase de 64 bits, el prefijo para estas direcciones es siempre FE80::/64. Un router IPv6 nunca envía tráfico de enlace local fuera del enlace.

- Direcciones de Sitio Local: Las direcciones de sitio local son equivalentes al espacio de direcciones privadas de IPv4 (10.0.0.0/8, 172.16.0.0/12, y 192.168.0.0/16). Por ejemplo, las intranets privadas que no poseen una

conexión directa y ruteable al Internet IPv6 pueden usar las direcciones de sitio local sin tener conflictos con las direcciones globales unicast. Las direcciones de sitio local no son accesibles desde otros sitios, y los routers no deben enviar tráfico de sitio local fuera del sitio. Dichas direcciones pueden ser usadas junto a las direcciones globales unicast. El ámbito de una dirección de sitio local es el sitio. Un sitio es la red de la organización ó una porción de dicha red, que tiene una localización geográfica definida, por ejemplo una oficina ó un campus.

A diferencia de las direcciones de enlace local, las direcciones de sitio local no son configuradas automáticamente y deben ser asignadas por medio de procesos de configuración de direcciones.

Figura 2.18: Dirección de sitio local

Los 10 primeros bits están siempre fijados para direcciones de sitio local (FEC0::/10). Después de dichos bits existe un identificador de subred que provee 54 bits con los que se puede crear una infraestructura jerárquica y resumida de ruteo dentro del sitio. Después de este campo está un identificador de interfase de 64 bits que identifica una interfase específica dentro de la subred.

2.5.7 Direcciones Unicast Globales Agregables

De acuerdo al RFC 2374[24], las direcciones Unicast Globales Agregables equivalen a las direcciones IPv4 públicas y son globalmente ruteables. Estas son identificadas por el prefijo de formato 001.

El concepto de agregable en el direccionamiento es necesario para una mejor organización jerárquica del ruteo en las redes globales. Este tipo de formato está diseñado para soportar agregaciones basadas en proveedores y un nuevo tipo basado en intercambios y la combinación de ambos hace que el ruteo sea más eficiente.

En este tipo de direcciones los 64 bits más altos identifican la red, y los 64 más bajos el nodo.

Las direcciones unicast globales agregables están organizadas en tres niveles de jerarquía:

- Topología Pública (*Public Topology*): es el conjunto de proveedores e intercambiadores que proveen servicios públicos de tránsito Internet.
- Topología de Sitio (*Site Topology*): es local a un sitio específico u organización que no provee servicio público a nodos fuera del sitio.
- Identificador de Interfaz (*Interface Identifier*): identifica las interfases en los links.

Existen tres tipos de direcciones unicast globales agregables:

[24] RFC 2374: An IPv6 Aggregatable Global Unicast Address Format

- 6Bone: comienzan con 3ffe, y son utilizadas para pruebas
- 6to4: comienzan con 2002
- Asignadas por un proveedor: comienzan con 2001

2.5.7.1 Estructura de las direcciones unicast globales agregables

<table>
<tr><td>3</td><td>13</td><td>8</td><td>24</td><td>16</td><td>64</td></tr>
<tr><td>FP</td><td>TLA ID</td><td>RES</td><td>NLA ID</td><td>SLA ID</td><td>INTERFACE ID</td></tr>
<tr><td colspan="4">Topología Pública</td><td>Topología de Sitio</td><td>Identificador de Interfaz</td></tr>
<tr><td colspan="5">Interfaz de Red</td><td>Interfaz de Host</td></tr>
</table>

Figura 2.19: Estructura de direcciones unicast globales agregables

- **FP** Format Prefix (001)
- **TLA ID** Top-Level Aggregation Identifier
- **RES** Reservado para uso futuro
- **NLA ID** Next-Level Aggregation Identifier
- **SLA ID** Site-Level Aggregation Identifier
- **INTERFACE ID** Interface Identifier

2.5.7.2 Descripción de los Campos

- FP: Se utiliza para identificar direcciones unicast globales agregables y su valor es 001.

- TLA ID: Se encuentra en el nivel superior de la jerarquía de ruteo. Los routers situados en este nivel tienen en su tabla de ruteo una entrada para cada TLA ID activo y probablemente tendrán entradas adicionales que proveerán información de ruteo del identificador de TLA en el cual se encuentran

  Podrían tener otras entradas, para optimizar el ruteo, dependiendo de su topología, pero siempre pensando en minimizar el número de entradas adicionales de la tabla de ruteo.

  Este formato de direccionamiento soporta 8192 identificadores TLA. Pudiéndose incrementar este número, aumentando el número de bits de este campo al utilizar los bits del campo reservado o usando este formato para prefijos de formato adicionales.

  Actualmente hay dos tipos de prefijos TLA:

  - Los establecidos por el 6Bone, cuyos primeros 16 bits comienzan por 3ffe::/16. Aquí los Top Level Aggregators son llamados pseudo-TLA o pTLA, los cuales son asignados a través de un proceso definido por la comunidad del 6bone.
  - Los de asignación de producción temprana cuyos primeros 16 bits son 2001::/16. Aquí los Top Level Aggregators son llamados sub-TLA, los cuales son asignados a través de la International Regional Internet Registry (RIR) Process.
- RES: Este campo se reserva para uso futuro y debe ser cero. Este campo permite un crecimiento futuro de los campos TLA ID y NLA ID.
- NLA ID: Este campo es usado por organizaciones asignadas a un TLA, para crear una estructura jerárquica de direccionamiento, acorde con su propia red, y para identificar los sitios u organizaciones que dependen de ella.

Cada organización puede manejar el NLA que le fue asignado, de forma que, reserve una porción para un nuevo NLA y crear así una jerarquía de direccionamiento apropiada a su red. El resto de los bits se utilizan para los sitios a los cuales desea dar servicio. Esto se muestra en el siguiente esquema:

n	24-n bits	16	64
NLA1	SITE ID	SLA ID	INTERFACE ID

Figura 2.20: NLA ID

Las organizaciones a las que se les asigna un TLA ID reciben 24 bits para uso del NLA ID. Este espacio permite a cada organización proveer servicio aproximadamente a tantas organizaciones como el número total de direcciones IPv4 soportadas actualmente.

Las organizaciones que tienen asignado un TLA ID pueden soportar varios NLA ID en su propio espacio de Site ID. Además las organizaciones que reciben un NLA ID pueden usar su Site ID para soportar otros NLAs ID. Esto se muestra en el siguiente esquema:

n	24-n bits	16	64
NLA1	SITE ID	SLA ID	INTERFACE ID

m	24-n-m bits	16	64
NLA2	SITE ID	SLA ID	INTERFACE ID

O	24-n-m-o bits	16	64

NLA3	SITE ID	SLA ID	INTERFACE ID

Figura 2.21: Jerarquía del NLA

El diseño del espacio del NLA ID para un TLA específico, es dejado a la organización responsable de ese TLA ID. Mientras que el diseño del siguiente NLA ID es responsabilidad del NLA ID del nivel previo.

- SLA: Este campo es usado por organizaciones finales para crear su propia jerarquía local de direccionamiento e identificar subredes. Es análogo al concepto de subred de IPv4 excepto que cada organización tiene un número mayor de subredes. Este campo soporta 65.355 subredes individuales.

  La forma en que se maneje el campo SLA ID es responsabilidad de cada organización. El número de subredes soportadas en este formato de direccionamiento debería ser suficiente, salvo para organizaciones muy grandes. Las organizaciones que necesiten subredes adicionales podrán solicitar otros identificadores SLA.

n	16-n bits	64
SLA1	Subred	INTERFACE ID

m	16-n-m bits	64
SLA2	SITE ID	INTERFACE ID

Figura 2.22: Jerarquía del SLA

- INTERFACE ID: Este campo es usado para identificar las interfases en el enlace. Estos deben ser únicos en su ámbito. En muchos casos los identificadores de interfase están basados en la dirección de la capa de enlace (MAC). Los identificadores de interfase usados en las

direcciones unicast globales agregables necesitan ser de 64 bits de longitud y construidos con el formato IEEE EUI-64[25].

2.5.7.3 Direcciones Especiales

Las siguientes son direcciones IPv6 especiales:

- Dirección No Especificada: (0:0:0:0:0:0:0:0 ó ::) Es solamente utilizada para indicar la ausencia de una dirección. Es equivalente a la dirección 0.0.0.0 en IPv4. La dirección no especificada es muy utilizada como una dirección de origen para paquetes de ensayo que verifican la unicidad de una dirección tentativa. La dirección no especificada nunca es asignada a una interfase ó usada como dirección de destino.
- Dirección de Loopback: (0:0:0:0:0:0:0:1 ó ::1) Es utilizada para identificar una interfase de loopback, permitiendo a un nodo enviar paquetes a sí mismo. La dirección de loopback es equivalente a la dirección 127.0.0.1 utilizada en IPv4. Los paquetes direccionados a la dirección de loopback no deben ser enviados nunca en un enlace ó enviados por un router IPv6.

2.5.7.4 Direcciones de Compatibilidad

A fin de ayudar en la migración de IPv4 a IPv6 y en la coexistencia de ambos tipos de hosts, se definieron las siguientes direcciones:

[25] IEEE EUI: Formato para direcciones IPv6. Institute of Electrical and Electronics Engineers. Extended User Interface

- Dirección compatible con IPv4: La dirección compatible con IPv4 tiene la forma 0:0:0:0:0:0:w:x:y:z ó ::w:x:y:z (donde w:x:y:z es la representación decimal separada por puntos, de una dirección IPv4). Es usada por nodos IPv6/IPv4 que se comunican usando IPv6; dichos nodos trabajan a la vez con ambos protocolos. Cuando una dirección compatible con IPv4 es usada como un destino IPv6, el tráfico IPv6 es automáticamente encapsulado con una cabecera IPv4 y es enviado a su destino usando una infraestructura IPv4.
- Dirección mapeada a IPv4: Una dirección mapeada a IPv4 tiene la forma 0:0:0:0:0:FFFF:w:x:y:z ó ::FFFF:w:x:y:z. Es usada para representar un solo nodo IPv4 á un nodo IPv6. Dichas direcciones son usadas solamente para representación interna. Una dirección mapeada a IPv4 nunca debe ser utilizada como dirección de origen ó destino de un paquete IPv6.
- Dirección 6to4: La dirección 6to4 es usada para comunicar entre sí a dos nodos que manejen tanto IPv4 como IPv6 sobre una infraestructura de ruteo IPv4. La dirección 6to4 está formada por la combinación del prefijo 2002::/16 con los 32 bits de una dirección IPv4 pública de un nodo, formando un prefijo de 48 bits.

2.5.8 Direcciones Multicast

En IPv6 el tráfico multicast opera de la misma forma que lo hace en IPv4. Arbitrariamente los nodos IPv6 pueden transmitir tráfico multicast en direcciones multicast arbitrarias. De igual forma los nodos IPv6 pueden atender múltiples direcciones multicast al mismo tiempo. Los nodos pueden unirse o dejar un grupo multicast en cualquier momento.

Las direcciones multicast IPv6 tienen los primeros ocho bits con el valor 1111 1111. Una dirección IPv6 es fácil de categorizar ya que siempre empieza con el valor “FF”. Las direcciones multicast no pueden ser usadas como dirección de origen ó como destinos intermedios en una cabecera de ruteo.

Más allá de los ocho primeros bits, las direcciones multicast incluyen una estructura adicional para identificar sus banderas, ámbitos y grupo multicast, como se muestra a continuación:

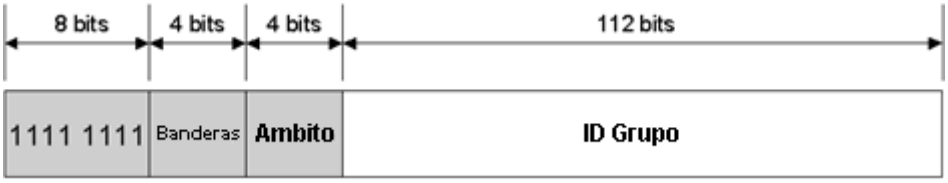


Figura 2.23: Dirección Multicast IPv6

Los campos de una dirección multicast son:

- Banderas: Como su nombre lo indica, este campo contiene banderas para el tratamiento de la dirección multicast. El tamaño de este campo es de 4 bits. Según el RFC 3513[26] la única bandera definida para este campo es la bandera Transitoria (T - Transient). La bandera T usa el bit de menor orden del campo de Banderas. Cuando tiene un valor de 0 indica que la dirección multicast es una dirección asignada permanentemente por parte de la IANA[27]. Cuando

[26] RFC 3513 : IPv6 Addressing Architecture
[27] IANA: Internet Assigned Numbers Authority

tiene un valor de 1 la bandera T indica que la dirección multicast es una dirección transitoria, es decir, no asignada permanentemente.

- Ámbito: Indica el ámbito de la red Internet IPv6 para el cual el tráfico multicast está dirigido. El tamaño de este campo es de 4 bits. Adicionalmente a la información proveída por los protocolos de ruteo multicast, los routers usan el ámbito multicast para determinar si el tráfico multicast puede ser transmitido. Los valores predominantes para el campo de Ámbito son 1 (ámbito de interfase local), 2 (ámbito de enlace local), y 5 (ámbito de sitio local).

  Por ejemplo, el tráfico con la dirección multicast FF02::2 tiene un ámbito de enlace local. Un router nunca envía este tráfico fuera del enlace local.

- ID Grupo: Identifica el grupo multicast y es único dentro del ámbito. El tamaño de este campo es de 112 bits. Los identificadores de grupos asignados permanentemente son independientes del ámbito. Las direcciones multicast desde FF01:: hasta FF0F:: son direcciones reservadas.

  Para identificar todos los nodos del ámbito de la interfase local y del enlace local, se definieron las siguientes direcciones:

  - FF01::1 ámbito de interfase local, dirección multicast de todos los nodos
  - FF02::1 ámbito de enlace local, dirección multicast de todos los nodos

  Para identificar todos los routers del ámbito de interfase local, del enlace local y del sito local, se definieron las siguientes direcciones:

  - FF01::2 ámbito de interfase local, dirección multicast de todos los nodos
  - FF02::2 ámbito de enlace local, dirección multicast de todos los nodos
  - FF05::2 ámbito de sitio local, dirección multicast de todos los nodos

Con los 112 bits es posible tener 2^{112} identificadores de grupos. Sin embargo, debido a la forma en que las direcciones multicast IPv6 son mapeadas a las direcciones MAC, se recomienda asignar el identificador de grupo con los 32 bits de menor orden de la dirección multicast y fijar el resto de bits en 0. Usando solamente dichos 32 bits, cada identificador de grupo mapea a una sola dirección MAC. La siguiente es la estructura modificada de la dirección multicast IPv6:

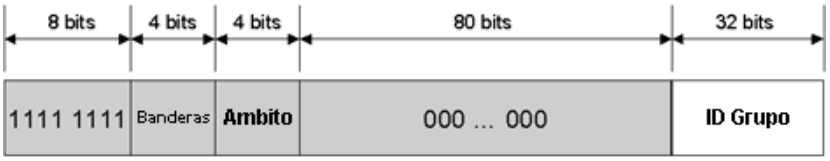


Figura 2.24: Dirección multicast IPv6 modificada, usando un ID Grupo de 32 bits

2.5.8.1 Direcciones Solicitadas por Nodos

Las direcciones solicitadas por nodos facilitan una eficiente búsqueda de nodos en la red durante la resolución de direcciones. En IPv4, el paquete de petición ARP[28] es enviado como broadcast a nivel MAC, perturbando a todos los nodos del segmento de red, incluyendo aquellos que no manejan IPv4. En cambio, IPv6 utiliza un mensaje de solicitud de vecindad para mejorar la resolución de direcciones. Sin embargo, en lugar de usar todos los nodos de la dirección multicast del enlace local como mensaje de destino de la solicitud de vecindad, el cual perturbará todos los nodos IPv6 del enlace local, se usa la dirección multicast solicitada por nodos. Dicha dirección está compuesta por el prefijo FF02::1:FF00:0/104 y los últimos 24 bits de la dirección IPv6 que está siendo resuelta, como se muestra a continuación:

[28] ARP: Address Resolution Protocol

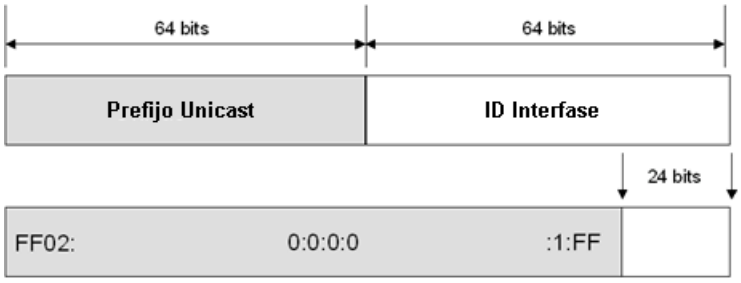


Figura 2.25: Dirección multicast solicitada por nodos

Por ejemplo, al nodo A se le asigna la dirección de enlace local de FE80::2AA:FF:FE28:9C5A y está también atendiendo en su correspondiente dirección multicast solicitada por nodos FF02::1:FF28:9C5A (el subrayado resalta la correspondencia en los seis últimos dígitos hexadecimales). El nodo B en el enlace local debe resolver la dirección de enlace local del nodo A FE80::2AA:FF:FE28:9C5A a su correspondiente dirección de capa de enlace. El nodo B envía un mensaje de solicitud de vecindad a la dirección multicast solicitada por nodos FF02::1:FF28:9C5A. Ya que el nodo A está atendiendo esta dirección multicast, se procesa el mensaje de solicitud de vecindad y envía un mensaje unicast de advertencia de vecindad a modo de respuesta.

El resultado de usar la dirección multicast solicitada por nodos es que la resolución de direcciones, algo común en un enlace, no requirieren usar un mecanismo que perturbe todos los nodos de la red. Al usar las direcciones solicitadas por nodos, muy pocos nodos son perturbados durante la resolución de direcciones. En la práctica, debido a la relación entre la dirección MAC, la interfase IPv6, y la dirección solicitada por nodos, esta dirección actúa como una pseudo-dirección unicast para resoluciones de direcciones más eficientes.

2.5.9 Direcciones Anycast

Una dirección anycast es asignada a múltiples interfases. Los paquetes direccionados a una dirección anycast son enviados por la infraestructura de ruteo hacia la interfaz más cercana a dicha dirección. A fin de facilitar las entregas, la infraestructura de ruteo debe saber cuáles son las interfases asignadas y su distancia, en términos de métricas de ruteo. Hoy en día las direcciones anycast son utilizadas solamente como direcciones de destino y son asignadas solamente a los routers. Las direcciones anycast son asignadas fuera del espacio de las direcciones unicast; el ámbito de una dirección unicast es el ámbito del tipo de dirección unicast desde la cual se asigna la dirección anycast.

La dirección anycast del router de subred es predefinida y necesariamente requerida. Es creada a partir del prefijo de subred para una interfaz dada. Para construir la dirección anycast del router de subred, los bits en el prefijo de subred son fijados con sus valores apropiados y los restantes toman el valor de 0. Todas las interfases que están ligadas a una subred son asignadas a la dirección anycast del router de dicha subred. El uso principal de la dirección anycast del router de subred es la comunicación con uno o varios routers ligados a una subred remota.

2.5.10 Direcciones IPv6 para un Host

Un host IPv4, con un adaptador de red, tiene una sola dirección IPv4 asignada a dicho adaptador. Sin embargo, un host IPv6 usualmente tiene

múltiples direcciones IPv6, aún con una sola interfase. Un host IPv6 tiene asignadas las siguientes direcciones unicast:

- Una dirección de enlace local, para cada interfase
- Una dirección unicast para cada interfase (puede ser una dirección de sitio local y una ó varias direcciones globales unicast
- La dirección de loopback (::1) para la interfase de loopback

Generalmente a los hosts IPv6 se les asigna lógicamente varias direcciones, ya que ellos tienen al menos dos direcciones con las cuales pueden recibir paquetes, una dirección de enlace local para el tráfico del enlace local, y una dirección ruteable de sitio local ó una dirección global. En varios casos un host tendrá las tres direcciones.

Adicionalmente, cada host procesa tráfico en las siguientes direcciones multicast:

- Todas las direcciones multicast de los nodos del ámbito de la interfase local
- Todas las direcciones multicast de los nodos del ámbito del enlace local
- La dirección solicitada por nodo de cada dirección unicast en cada interfase
- Las direcciones multicast de grupos adjuntos en cada interfase

2.5.11 Direcciones IPv6 para un Router

Un router IPv6 tiene asignadas las siguientes direcciones unicast:

- Una dirección de enlace local para cada interfase

- Una dirección unicast para cada interfase (puede ser una dirección de sitio local y una o varias direcciones globales unicast)
- Una dirección anycast del router de subred
- Direcciones anycast adicionales (opcional)
- La dirección de loopback (::1) para la interfase de loopback

Adicionalmente, cada router procesa tráfico en las siguientes direcciones multicast:

- Todas las direcciones multicast de los nodos del ámbito de la interfase local (FF01::1)
- Todas las direcciones multicast de los nodos del ámbito del enlace local (FF02::1)
- Todas las direcciones multicast de los routers del ámbito del enlace local (FF02::2)
- Todas las direcciones multicast de los routers del ámbito del sitio local (FF05::2)
- La dirección solicitada por nodo de cada dirección unicast de cada interfase
- Las direcciones multicast de grupos adjuntos en cada interfase

2.5.12 Conceptos de Direccionamiento en IPv4 e IPv6

Tabla 2.6: Conceptos de Direccionamiento en IPv4 y sus equivalentes en IPv6

Dirección IPv4	**Dirección IPv6**
Clases de Direcciones en Internet	No aplicable en IPv6
Direcciones Multicast (224.0.0.0/4)	Direcciones Multicast IPv6 (FF00::/8)

Direcciones Broadcast	No aplicable en IPv6
La dirección no especificada es 0.0.0.0	La dirección no especificada es ::
La dirección de loopback es 127.0.0.1	La dirección de loopback es ::1
Direcciones IP Públicas	Direcciones Globales Unicast
Direcciones IP Privadas (10.0.0.0/8, 172.16.0.0/12, y 192.168.0.0/16)	Direcciones de Sitio Local (FEC0::/10)
Direcciones Autoconfiguradas(169.254.0.0/16)	Direcciones de Enlace Local (FE80::/64)
Representación de texto: Notación decimal separada por puntos	Representación de texto: Formato hexadecimal separado por dos puntos, con supresión de ceros restantes y compactación de ceros. Las direcciones compatibles IPv4 son expresadas en notación decimal separada por puntos.
Representación de bits de subred: Máscara de subred en notación decimal separada por puntos ó longitud de prefijo.	Representación de bits de subred: Solamente notación de longitud de prefijo.
Resolución de nombres DNS: Registro de recursos de direcciones de hosts IPv4 (A)	Resolución de nombres DNS: Registro de recursos de direcciones de hosts IPv6 (AAAA), actualmente basados en servidores IPv4
Resolución inversa DNS: dominio IN-ADDR.ARPA	Resolución inversa DNS: Dominio IP6.ARPA

2.5.13 ICMPv6

El protocolo de Mensajes de Control de Internet (ICMP), ha sido actualizado para su uso bajo IPv6 y se le ha asignado el valor de 58 en el campo de “siguiente cabecera” para saber que es un ICMPv6. Este protocolo es parte integral de IPv6 y debe ser totalmente incorporado a cualquier implementación de nodo IPv6.

ICMPv6 es empleado por IPv6 para reportar errores que se encuentran durante el procesado de los paquetes, así como para la realización de otras funciones relativas a la capa Internet, como son las de diagnósticos ping.

El formato genérico de los mensajes ICMPv6 es el siguiente:

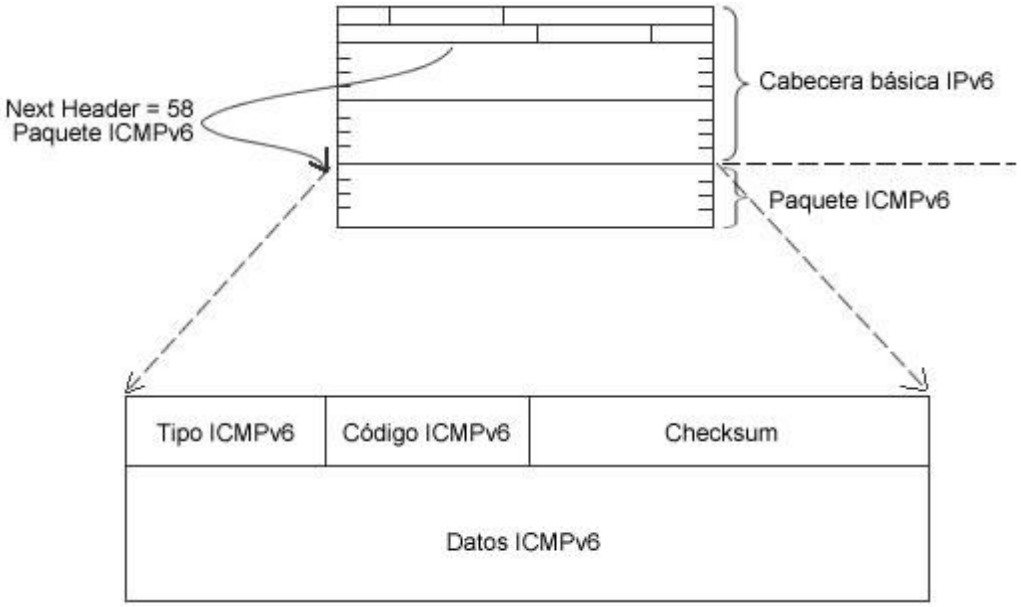


Figura 2.26: Formato del paquete ICMPv6

- Tipo: Indica el tipo de mensaje, y su valor determina el formato del resto de la cabecera
- Código: Depende del tipo de mensaje, y se emplea para crear un nivel adicional de jerarquía para la clasificación del mensaje.
- Checksum: Código de redundancia. Permite detectar errores en el mensaje ICMPv6.

2.5.13.1 Tipos de ICMPv6

Los mensajes de ICMP, se han dividido en 2 clases, los que comunican errores, y los que piden/dan información sobre un nodo. Los mensajes de error tienen cero en el bit de mayor peso del campo “tipo”, por lo que sus valores se sitúan entre 0 y 127 y los valores de los mensajes informativos oscilan entre 128 y 255.

2.5.13.2 Tipos de información ICMPv6

Los mensajes de información, pueden ser del tipo:

- Echo Request (Type 128): Un nodo puede enviar un ICMP Echo Request (más conocido como ping) para saber el tiempo de respuesta de otro host. La recepción de ICMP Echo Request, debe ser comunicada a la capa superior de transporte.
- Echo Reply (Type 129): El ICMP Echo Reply es enviado como respuesta a un ICMP Echo Request, y debe ser trasportado al proceso que origino el ICMP Echo Request.

2.5.13.3 Tipos de error ICMPv6

- Destination Unreachable (Type 1): Un ICMP Destination Unreachable es enviado por un router, o por cualquier nodo, para informar de la imposibilidad de que un paquete llegue a su destino. No se deberían enviar estos mensajes si son ocasionados por problemas de congestión de la red. Un nodo que ha recibido un ICMPv6 Destination Unreachable, debe comunicarlo a la capa superior del proceso.
- Packet Too Big (Type 2): Un ICMP Packet Too Big es enviado cuando el tamaño máximo de un paquete es superior a la MTU de la interfaz de red al que se ha enviado. También es enviado por un router si el siguiente salto tiene un MTU inferior al tamaño del paquete. Este ICMP puede ser usado para saber el MTU de una ruta.
- Time Exceeded (Type 3): Este mensaje se emite cuando se ha llegado al límite de saltos establecido para el envío de un paquete. Si un host no puede ensamblar un paquete en un tiempo dado se descartarán todos los fragmentos recibidos y se enviará un mensaje de este tipo.
- Parameter Problem (Type 4): Si al procesar un paquete, un nodo IPv6 encuentra un error en uno de los parámetros de sus campos, enviará un ICMP Parameter Problem informando al destino de la situación del error en el paquete.

Tabla 2.7: Resumen de los mensajes ICMPv6

Tipo	Descripción y Códigos

<table>
<tr><td rowspan="7">1</td><td colspan="2">Destino no alcanzable</td></tr>
<tr><td>Código</td><td>Descripción</td></tr>
<tr><td>0</td><td>Sin ruta hacia el destino</td></tr>
<tr><td>1</td><td>Comunicación prohibida administrativamente</td></tr>
<tr><td>2</td><td>Sin asignar</td></tr>
<tr><td>3</td><td>Dirección no alcanzable</td></tr>
<tr><td>4</td><td>Puerto no alcanzable</td></tr>
<tr><td>2</td><td colspan="2">Paquete demasiado grande</td></tr>
<tr><td rowspan="4">3</td><td colspan="2">Tiempo excedido</td></tr>
<tr><td>Código</td><td>Descripción</td></tr>
<tr><td>0</td><td>Límite de saltos excedido</td></tr>
<tr><td>1</td><td>Tiempo de desfragmentación excedido</td></tr>
<tr><td rowspan="5">4</td><td colspan="2">Problemas de parámetros</td></tr>
<tr><td>Código</td><td>Descripción</td></tr>
<tr><td>0</td><td>Campo erróneo en cabecera</td></tr>
<tr><td>1</td><td>Tipos de “cabecera siguiente” desconocida</td></tr>
<tr><td>2</td><td>Opción IPv6 desconocida</td></tr>
<tr><td colspan="3">Mensajes informativos ICMPv6</td></tr>
<tr><td>Tipo</td><td colspan="2">Descripción</td></tr>
<tr><td>128</td><td colspan="2">Solicitud de eco</td></tr>
<tr><td>129</td><td colspan="2">Respuesta de eco</td></tr>
</table>

2.5.14 Protocolos de ruteo IPv6

Básicamente se adoptan los mismos protocolos de encaminamiento que los existentes en las redes IPv4: RIP, OSPF[29], BGP.

2.5.14.1 RIPng

[29] OSPF: Open Shortest Path-First Protocol

RIPng es un protocolo diseñado para pequeñas redes y por tanto se incluye en el grupo de protocolos de pasarela interior (IGP[30]). Emplea el algoritmo vector distancia, y se basa en el intercambio de información entre routers de forma que puedan calcular las rutas más adecuadas, de forma automática.

RIPng solo puede ser implementado en routers, donde requerirá como información fundamental la métrica o número de saltos que un paquete ha de emplear para llegar a determinado destino. Además de la métrica, cada red tendrá un prefijo de dirección destino y la longitud del propio prefijo.

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

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

Al igual que en IPv4, el inconveniente de RIPng sigue siendo su orientación a pequeñas redes y el hecho de que su métrica es fija y no puede variar en función de las circunstancias de un ambiente de producción.

2.5.14.2 BGP4+

[30] IGP: Interior Gateway Protocol

El protocolo BGP fue diseñado para la interconexión de sistemas autónomos. Frecuentemente se emplea en grandes ambientes de comunicaciones y para la conexión entre proveedores de servicios (como ISPs). Su principal función es, por tanto, el intercambio de información de disponibilidad o alcance entre varios sistemas BGP, incluyendo información de los sistemas autónomos que contienen; permitiendo así construir las rutas más adecuadas y evitar bucles de tráfico. Los equipos que utilizan el protocolo BGP solamente informan a los equipos que se conectan a ellos, acerca de las rutas que emplean, es decir, es una estrategia de “salto a salto”.

2.5.15 Neighbor Discovery

Es el protocolo de descubrimiento de vecinos, y es el mecanismo por el cual un nodo que se incorpora a una red descubre la presencia de otros en su mismo enlace, a fin de determinar sus direcciones en la capa de enlace; además permite localizar los routers, y mantener la información de conectividad acerca de las rutas a los vecinos activos. Este mecanismo además permite descubrir las características del medio por donde van a circular los paquetes enviados por el host.

El protocolo Neighbor Discovery define entre otros los siguientes mecanismos: descubrimiento de routers, prefijos y parámetros, autoconfiguración de direcciones, resolución de direcciones, determinación del siguiente salto, detección de nodos no alcanzables, detección de direcciones duplicadas o

cambios de direcciones, redirección, balanceo de carga entrante y direcciones anycast.

El protocolo Neighbor Discovery define cinco tipos de paquetes ICMPv6:

- Router Discovery (Descubrimiento de router): Es generado cuando una interfaz se activa, y solicita información sobre los routers activos. Este mensaje comunica al host y a otros routers la existencia de un nuevo router, o la permanencia ó eliminación de los actuales.
- Router Advertisement (Anuncio de router): Se genera periódicamente por los routers, a través de multicast, a fin de informar de su presencia así como de otros parámetros de enlace.
- Neighbor Solicitation (Solicitud de Vecino): Es generado por los nodos para determinar la dirección en la capa de enlace de sus vecinos, o para verificar que el nodo vecino sigue activo, así como para detectar direcciones duplicadas.
- Neighbor Advertisement (Anuncio de Vecino): Es generado por los nodos como respuesta a la solicitud de vecino, o bien para indicar cambios de direcciones en la capa de enlace.
- Redirect (Redirección): Se genera en los routers para informar a los host de un salto más eficiente para llegar a un determinado destino.

2.5.15.1 Ventajas del protocolo ND

Las principales ventajas del protocolo Neighbor Discovery son:

- Permite manejar con mayor eficacia la información de ruteo de las redes que es almacenada por los equipos de comunicación.
- Disminuye el intercambio de paquetes de información entre equipos, y el uso de protocolos de enrutamiento.
- Gracias al anuncio de routers permite la autoconfiguración de direcciones para los equipos que se integran a las redes.
- Utiliza la detección de vecinos no alcanzables para mejorar la robustez en la entrega de paquetes frente a fallos en routers, enlaces, nodos que cambian sus direcciones, nodos móviles, etc.

2.5.16 Autoconfiguración en IPv6

El proceso de autoconfiguración incluye la creación de una dirección de enlace local y la verificación de su unicidad en un enlace, determinando qué información debe ser autoconfigurable (dirección, otra información), y en el caso de direcciones si serán obtenidas mediante el proceso sin estados, con estados o ambos.

IPv6 define mecanismos de autoconfiguración de direcciones, tanto con estado (stateful) como sin estado (stateless). La autoconfiguración stateless requiere que no se haga ninguna configuración manual de hosts, una mínima configuración en los routers (si es necesaria), y no necesita de servidores adicionales. Dicho mecanismo permite que una máquina genere su propia dirección utilizando una combinación de información disponible localmente y mensajes anunciados por los routers. Las direcciones se conforman utilizando los prefijos anunciados por los routers, que identifican a una o varias subredes asociadas a un enlace, y un identificador de interfase generado por los hosts, que distingue solamente a una interfase de la red. Dado el caso de que no existan routers dentro de la red, un host solamente puede generar direcciones de enlace local, las cuales son suficientes para permitir la comunicación entre nodos del mismo enlace.

En cambio en la autoconfiguración stateful son los servidores quienes proveen las direcciones para los hosts de la red, así como información y parámetros de configuración. Dichos servidores mantienen una base de datos con

el registro de cada dirección asignada a cada host de la red. Ambos mecanismos de autoconfiguración son complementarios, es decir, pueden trabajar en conjunto sin ningún problema. Para citar un ejemplo podemos mencionar un host que usa autoconfiguración stateless para obtener sus direcciones, mientras que usa la autoconfiguración stateful para obtener la información y parámetros de configuración.

La aproximación stateless se emplea cuando en un sitio no importa la dirección exacta que se asigna a un host, sino el hecho de que la misma sea única y correctamente ruteable. Este mecanismo es utilizado cuando un sitio necesita de un estricto control de la asignación de direcciones. Ambas autoconfiguraciones pueden ser utilizadas simultáneamente, pero es el administrador del sitio quien especificará cuál de ellas usará, por medio de la configuración de los campos apropiados en los mensajes de anuncio de router.

Las direcciones IPv6 son asignadas a una interfase por un tiempo determinado, el mismo que puede ser infinito dependiendo de las necesidades del host. Cada dirección tiene un tiempo de vida, el cual indica el periodo en el cual una dirección está asociada con una interfase. Una vez que este periodo culmina dicha asociación queda inválida y la dirección puede ser reasignada a alguna otra interfase.

A fin de asegurar que todas las direcciones configuradas sean únicas en un enlace, los nodos ejecutan un algoritmo denominado “detección de direcciones duplicadas” en cada dirección antes de asignarla a una interfase. Dicho algoritmo

se ejecuta tanto en direcciones asignadas por medio de la autoconfiguración stateless como stateful.

Cabe recalcar que el proceso de autoconfiguración descrito solamente se aplica en hosts más no en routers, debido a que se utiliza información anunciada por routers.

2.5.16.1 Autoconfiguración Stateless

El procedimiento de autoconfiguración stateless (sin intervención o descubrimiento automático), ha sido diseñado con las siguientes premisas:

- Evitar la configuración manual de dispositivos antes de su conexión a la red.
  - Se requiere, en consecuencia, un mecanismo que permita a los hosts obtener o crear direcciones únicas para cada una de sus interfaces, asumiendo que cada interfaz puede proporcionar un identificador único para si mismo (identificador de interfaz).
  - En el caso más simple, el identificador de interfaz consiste en la dirección de la capa de enlace, de dicha interfaz.
  - El identificador de interfaz puede ser combinado con un prefijo, para formar la dirección

- Las máquinas que integran una red local no deberían requerir la presencia de un servidor stateful o router, como requisito para comunicarse.
  - Para obtener, en este caso, características plug & play, empleamos las direcciones de enlace local, dado que tienen un prefijo perfectamente

conocido que identifica el único enlace compartido, al que se conectan todos los nodos.

- Cada dispositivo forma su dirección de enlace local anteponiendo el prefijo de enlace local a su identificador de interfaz.

- En el caso de redes o sitios grandes, con múltiples subredes y routers, tampoco se requiere la presencia de un servidor de configuración de direcciones stateful, ya que los hosts han de determinar, para generar sus direcciones globales o de enlace local, los prefijos que identifican las subredes a las que se conectan. Los routers generan mensajes periódicos de anunciación 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, por ejemplo, cuando se desea cambiar de proveedor de servicios.
  - La renumeración se logra al permitir que una misma interfaz pueda tener varias direcciones, que recibe en préstamo.
  - El tiempo del préstamo es el mecanismo por el que se renuevan las direcciones, al expirar los plazos para las viejas, sin que se conceda una prórroga.
  - Al poder disponer de varias direcciones simultáneamente, permite que la transición no sea interrumpida, permitiendo que ambas, la vieja y la nueva dirección den continuidad a la comunicación durante el periodo de transición.

- Solo es posible utilizar este mecanismo en enlaces capaces de funciones multicast, y comienza, por tanto, cuando es iniciada o activada una interfaz que permite multicast.

- Los administradores de sistemas necesitan la habilidad de especificar que mecanismos (stateless, stateful, o ambos), deben ser usados. Los mensajes de anunciación de los routers incluyen indicadores para esta función.

Los pasos básicos para la autoconfiguración, una vez la interfaz ha sido activada, serían:

- Se genera la dirección tentativa de enlace local.
- Verificar que dicha dirección tentativa puede ser asignada (no esta duplicada en el mismo enlace).
- Si esta duplicada, la autoconfiguración se detiene, y se requiere un procedimiento manual (por ejemplo, usando otro identificador de interfaz)
- Si no está duplicada, la conectividad a nivel IP se ha logrado, al asignarse definitivamente dicha dirección tentativa a la interfaz en cuestión
- Si se trata de un host, se interroga a los posibles routers para indicar al host lo que debe de hacer a continuación
- Si no hay routers, se invoca el procedimiento de autoconfiguración stateful
- Si hay routers, estos contestarán indicando fundamentalmente, como obtener las direcciones si se ha de utilizar el mecanismo stateful, u otra información, como tiempos de vida, etc.

2.5.16.2 Autoconfiguración Stateful – DHCPv6

DHCP para IPv6 es un protocolo UDP cliente/servidor, diseñado para reducir el costo de gestión de nodos IPv6 en entornos donde los administradores precisan un control sobre la asignación de los recursos de la red, superior a los facilitados por el mecanismo de configuración stateless.

Ambos mecanismos pueden usarse de forma concurrente para reducir el costo de propiedad y administración de la red.

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

Además, DHCP ha sido diseñado para ser fácilmente extensible con nuevos parámetros de configuración, a través de extensiones que incorporan esta nueva información. A continuación se enumeran algunas de las características de DHCPv6:

- 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 autoconfiguración stateless

- 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 coexiste con nodos configurados estáticamente, así como con implementaciones existentes en la red.
- Los clientes DHCP pueden operar en enlaces donde no hay routers 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, está basados en el soporte inherente del formato de direccionamiento y autoconfiguración IPv6:

- Las direcciones 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.
- 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 stateful ha de coexistir e integrarse con la stateless, soportando la detección de direcciones duplicadas y los dos tiempos de vida de IPv6, para facilitar la renumeración automática de direcciones y su gestión.
- Se soportan múltiples direcciones por cada interfaz.
- Algunas opciones DHCPv4 ya no son precisas, debido a que los parámetros de configuración se obtienen a través de ND o del producto de localización de servicios.

De esta forma, se soportan las siguientes funciones nuevas:

- Configuración de actualizaciones dinámicas de DNS
- Desaprobación de direcciones, para renumeración dinámica.
- Autenticación.
- Los clientes pueden pedir múltiples direcciones IP
- Las direcciones pueden ser reclamadas mediante el mensaje de iniciar reconfiguración.
- Integración entre autoconfiguración de direcciones stateless y stateful

2.5.16.3 Renumeración

Ya se ha descrito el mecanismo básico de renumeración basado en el préstamo o alquiler de direcciones, cuando se dice preferida y desaprobada, y en el tiempo de vida de las mismas.

En cualquier caso, se puede describir el mecanismo de forma sencilla, como consistente en disminuir el tiempo de vida del prefijo en los paquetes de

anunciación del router, de forma que las direcciones pasen a ser desaprobadas, frente a las nuevas, que pasan a ser preferidas.

2.6 Comparación entre IPv4 e IPv6

- No hay direcciones broadcast. Su funcionalidad le corresponde a las direcciones multicast.
- Los campos de las direcciones reciben nombres específicos.
- El prefijo permite conocer donde está conectada una determinada dirección, es decir, su ruta de encaminamiento.
- Cualquier campo puede contener solo ceros o solo unos, salvo que se indique explícitamente lo contrario.
- Las direcciones IPv6, indistintamente de su tipo (unicast, anycast o multicast), son asignadas a interfases y no a nodos. Dado que cada interfaz pertenece a un único nodo, cualquiera de las direcciones unicast de las interfaces del nodo puede ser empleado para referirse a dicho nodo.
- Todas las interfases deben tener, al menos, una dirección unicast de enlace local
- Una única interfaz puede tener también varias direcciones IPv6 de cualquier tipo (unicast, anycast o multicast)
- Una misma dirección o conjunto de direcciones unicast pueden ser asignados a múltiples interfaces físicas, siempre que la implementación trate dichas interfaces, desde el punto de vista de Internet, como única, lo que permite balanceo de carga entre múltiples dispositivos.

- Al igual que en IPv4, se asocia un prefijo de subred con un enlace, y se pueden asociar múltiples prefijos de subred a un mismo enlace.
- IPv6 soporta un mayor espacio de direcciones. El tamaño de las direcciones cambia de 32 bits a 128 bits, para soportar: más niveles de jerarquías de direccionamiento y más nodos direccionables.
- Simplificación del formato de la cabecera. Algunos campos de la cabecera IPv4 han sido eliminados ó son opcionales.
- Los paquetes IP son más eficientes y extensibles, sin que haya fragmentación en los routers. Debido a que son alineados a 64 bits y poseen una cabecera de longitud fija, más simple, se agiliza su procesamiento por parte del router.
- Se implementa una mayor seguridad pues el soporte para IPSec es un requerimiento del protocolo IPv6.
- Los paquetes IPv6 manejan etiquetas de flujo, las cuales pueden ser usadas por un nodo origen para etiquetar paquetes pertenecientes a un flujo de tráfico particular, que requiere manejo especial por los routers IPv6, tal como calidad de servicio no por defecto o servicios de tiempo real.
- La autoconfiguración de direcciones es más simple, especialmente en las direcciones globales unicast agregables, donde los 64 bits superiores son seteados por un mensaje desde el router y los 64 bits más bajos son seteados con la dirección MAC.
- La longitud del prefijo de red no depende del número de los hosts y por tanto la asignación es más simple.
- IPv6 posee características de renumeración y multihomming, lo que facilita un posible cambio de proveedor de servicios.

- El protocolo IPv6 posee mejores características de movilidad, incluyendo la posibilidad de que un nodo mantenga la misma dirección IP sin importar su ubicación geográfica.
- En IPv6 existe un ruteo más eficiente en el backbone de la red, debido a su jerarquía de direccionamiento.
- IPv6 presenta mejoras en la calidad de servicio (QoS) y clase de servicio (CoS)

2.7 Mecanismos de transición para Hosts y Routers IPv6

2.7.1 Introducción

La exitosa adopción de una nueva tecnología depende de su fácil integración con la infraestructura existente, sin caer en una alteración significativa de los servicios. El Internet está conformado por cientos de miles de redes IPv4 y millones de nodos IPv4. El reto consiste en hacer de la integración y la transición entre tecnologías lo más transparente posible para los usuarios finales.

El RFC 2893[31] define un conjunto de mecanismos que pueden ser usados por hosts y routers IPv6 para ser compatibles con sus homólogos en IPv4. Los mecanismos definidos son los siguientes:

- Doble capa IP ó Dual Stack. Esta técnica provee un soporte completo para ambos protocolos IPv4 e IPv6 tanto en hosts como en routers.

[31] RFC 2893: Transition Mechanisms for IPv6 Hosts and Routers

- Configuración de tunneling IPv6 sobre IPv4, es decir que se configura un túnel punto a punto haciendo que los paquetes IPv6 se encapsulen dentro de las cabeceras IPv4 para llevarlos sobre la infraestructura IPv4.
- Direcciones IPv4 compatibles con IPv6, es decir que en el formato de una dirección IPv6 se usa una dirección IPv4 embebida.
- Tunneling Automático de IPv6 sobre IPv4, este mecanismo utiliza la dirección IPv4 compatible para el túnel automático de paquetes IPv6 sobre redes IPv4.

2.7.2 Dual Stack (Doble Pila)

El uso de un backbone dual stack es una técnica básica para el ruteo de IPv4 e IPv6, y obviamente requiere de dispositivos de red que manejen las pilas de ambos protocolos. Los sistemas dual stack permiten a las aplicaciones migrar, una a la vez, de una comunicación IPv4 a una IPv6. Las aplicaciones que no son actualizadas para soportar la pila IPv6 no pueden coexistir con las aplicaciones actualizadas dentro del mismo sistema.

Los nodos IPv6 que presentan esta característica son denominados "IPv6/IPv4". Este tipo de nodo tiene la capacidad de enviar y recibir ambos tipos de paquetes, es decir operan directamente tanto con nodos y paquetes IPv4 e IPv6.

Como se muestra a continuación las aplicaciones nuevas y las actualizadas hacen uso de las pilas de los protocolos IPv4 e IPv6. Una nueva API[32] ha sido definida para soportar direcciones IPv4 e IPv6 y peticiones DNS. Una aplicación

[32] API: Application Programming Interface – Interfase de Programación de Aplicaciones

puede ser actualizada a la nueva API y seguir utilizando solamente la pila del protocolo IPv4.

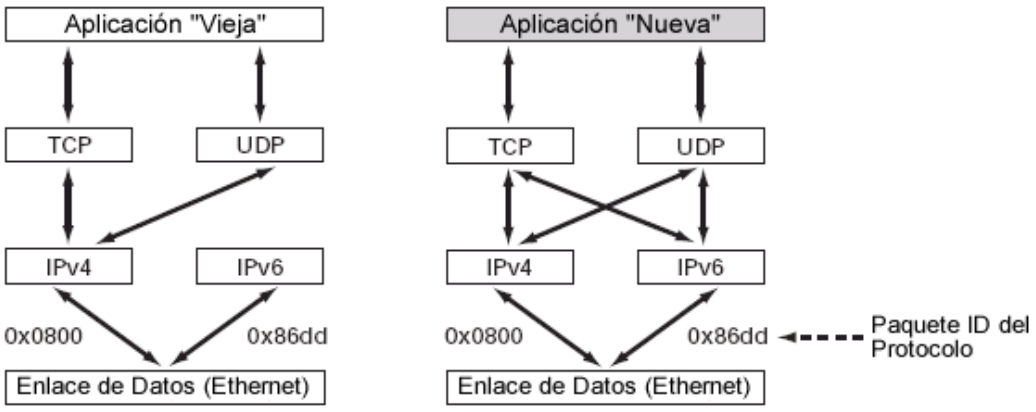


Figura 2.27: Pila Doble (Dual Stack) IPv4-IPv6

Debido al uso de backbones dual stack todos los routers de una red necesitan ser actualizados para soportar una comunicación entre ambos protocolos, tomando en cuenta el tipo de tráfico IP y los requerimientos de la comunicación.

Actualmente el ruteo dual stack es una estrategia muy utilizada en el desarrollo de infraestructuras de red que manejan aplicaciones que trabajan con protocolos IPv4 e IPv6. Sin embargo, existen ciertas limitaciones para el uso de esta alternativa; a más de la obvia necesidad de actualizar todos los routers de una red se suman problemas como que todos los routers requieren la definición de un doble esquema de direccionamiento, además se necesita una doble administración de los protocolos de ruteo, y por último, que los routers deben ser

configurados con suficiente memoria para almacenar las tablas de ruteo de ambos protocolos.

2.7.3 Mecanismos comunes de tunneling

La infraestructura IPv6 será implementada paulatinamente con el pasar del tiempo y mientras esto sucede, la infraestructura IPv4 existente puede agregar a su funcionalidad la capacidad de llevar tráfico IPv6.

Los host y routers dual stack pueden transportar datagramas IPv6 por medio de un túnel que trabaje sobre regiones con topologías IPv4 mediante el encapsulamiento de los mismos dentro de paquetes IPv4. A continuación se especifican los detalles de las principales técnicas a fin de tener una base conceptual que permita elegir el mecanismo adecuado para la situación específica de una red:

- Túneles IPv6 configurados manualmente
- IPv6 sobre túneles GRE IPv4
- Túneles automáticos compatibles con IPv4
- Túneles automáticos 6to4

La principal diferencia entre los túneles automáticos y configurados es la forma como se determina la dirección del punto final del túnel. Los fundamentos de ambos mecanismos son similares:

- El punto inicial del túnel (nodo encapsulador) crea una cabecera IPv4 y transmite el paquete encapsulado.
- El punto final del túnel (nodo desencapsulador) recibe el paquete encapsulado, reensambla el paquete si es necesario, remueve la cabecera IPv4, actualiza la cabecera IPv6 y procesa el paquete IPv6 recibido.
- El nodo encapsulador puede tener la necesidad de almacenar cierto tipo de información para cada túnel a fin de procesar los paquetes IPv6 enviados por medio del túnel. Dado que el número de túneles utilizados puede ser alto, dicha información puede ser almacenada temporalmente y luego descartada.

2.7.3.1 Encapsulamiento

El encapsulamiento de un datagrama IPv6 en IPv4 se muestra a continuación:

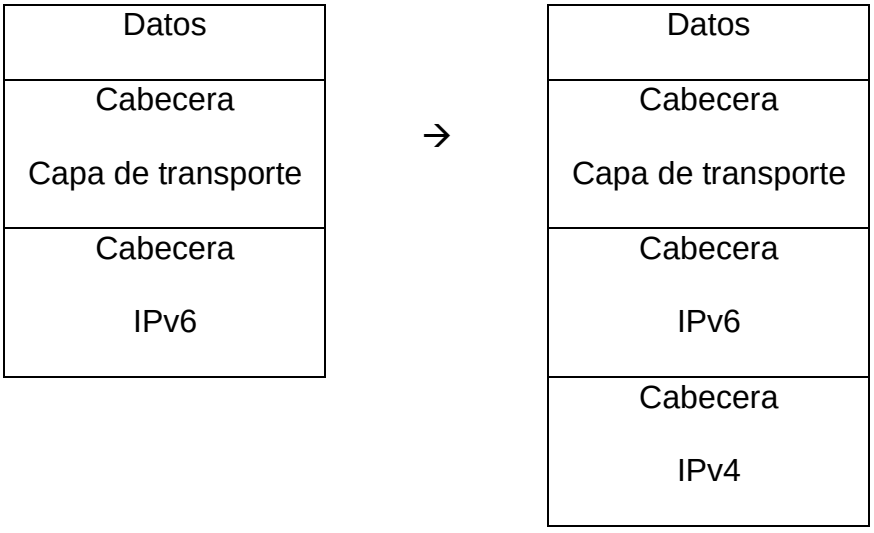


Figura 2.28: Encapsulamiento de un datagrama IPv6 en IPv4

Además de agregar una cabecera IPv4, el nodo encapsulador también tiene que manejar algunos aspectos más complejos:

- Determinar cuando fragmentar y cuando reportar al nodo emisor un error ICMP acerca de un paquete demasiado grande.
- Como reflejar un error ICMPv4 desde los routers y regresarlo al nodo emisor como un error ICMPv6

2.7.3.2 Límite de saltos

Los túneles IPv6 sobre IPv4 son diseñados como túneles de un único salto. Es decir, el límite de saltos IPv6 es decrementado en 1 cuando un paquete IPv6 atraviesa el túnel. El modelo de un único salto sirve para ocultar la existencia de un túnel. Este es transparente para los usuarios de la red y no es detectado por herramientas de diagnóstico de la red como el traceroute.

El TTL de la cabecera IPv4 encapsulada es seleccionado de diferente manera en las implementaciones, las cuales deben proveer mecanismos para configurar el TTL IPv4.

2.7.3.3 Construcción de la cabecera IPv4

En el momento del encapsulamiento de un paquete IPv6 en datagramas IPv4, los campos de la cabecera IPv4 son modificados como sigue:

- Versión: 4
- Longitud de la cabecera IP en 32 bits: 5 (no hay opciones IPv4 en la cabecera de encapsulamiento)
- Tipo de servicio: 0
- Longitud Total: Longitud de carga útil desde la cabecera IPv6 más la longitud de las cabeceras IPv6 e IPv4
- Identificación: Generada únicamente como por algunos paquetes IPv4 trasmitidos por el sistema

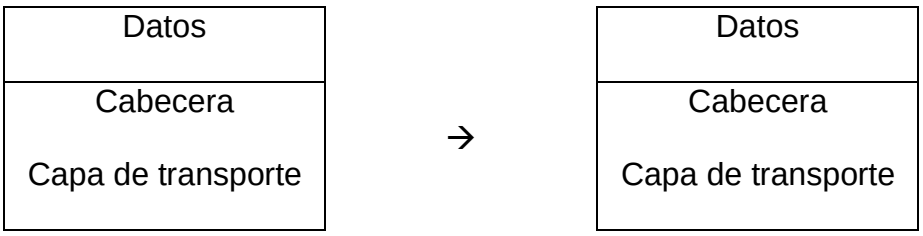
- Banderas: El bit de No Fragmentación (DF) seteado a 0. El bit de Más Fragmentos (MF) seteado si se necesitan más fragmentaciones
- Desplazamiento de fragmento: Es seteado si se necesitan más fragmentaciones
- Tiempo de vida: Seteado en modos de implementación específica
- Protocolo: 41. Número asignado por IPv6 para describir el tipo de carga útil
- Cabecera Checksum: Valor de checksum calculado en la cabecera IPv4
- Dirección origen: Dirección IPv4 de la interfase saliente del nodo encapsulador
- Dirección destino: Dirección IPv4 del final del túnel

2.7.3.4 Desencapsulamiento

Cuando un host o un router dual stack recibe un datagrama IPv4 que está direccionado a una de sus direcciones IPv4, y el valor del campo protocolo es 41, este lo reensambla si el paquete es fragmentado en el nivel IPv4, entonces la cabecera IPv4 es removida y se envía el datagrama IPv6.

El nodo desencapsulador debe ser capaz de reensamblar un paquete IPv4 de 1300 bytes (1280 bytes más la cabecera IPv4 que en este caso siempre será de 20 bytes)

La desencapsulación se muestra como sigue:



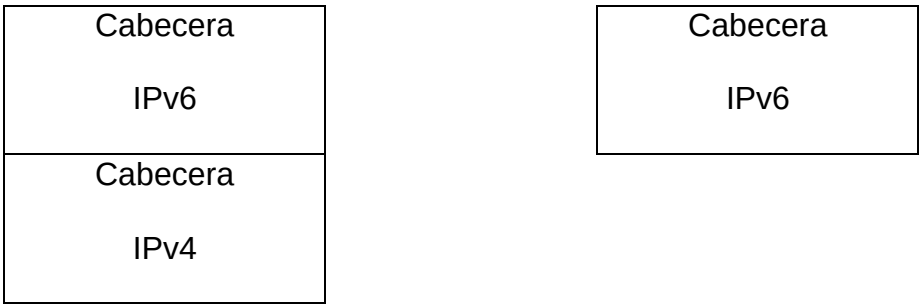


Figura 2.29: Desencapsulamiento de un datagrama IPv6 en IPv4

Cuando se desencapsula un paquete, la cabecera IPv6 no es modificada. Si el paquete es reenviado posteriormente, el campo límite de saltos es decrementado en uno.

Como parte de la desencapsulación el nodo debe descartar tácitamente un paquete cuando cumple las siguientes condiciones:

- Para IPv4, cuando la dirección de origen IPv4 no es válida como en el caso de una dirección multicast, broadcast, 0.0.0.0 y 127.0.0.1. La cabecera de encapsulamiento IPv4 es descartada.
- Para IPv6, cuando la dirección de origen IPv6 es inválida; esto incluye direcciones IPv6 multicast, una dirección no especificada y la dirección de loopback. Adicionalmente cuando esta dirección es del tipo IPv4 compatible con IPv6 y la parte IPv4 es una dirección multicast, broadcast, 0.0.0.0 o 127.0.0.1.

El nodo desencapsulador realiza el reensamblamiento IPv4 antes de desencapsular el paquete IPv6. Todas las opciones IPv6 son conservadas aún sí el paquete IPv4 encapsulado es fragmentado.

Después de que el paquete IPv6 es desencapsulado, es procesado casi de la misma forma que cualquier paquete IPv6 recibido. La única diferencia que existe es que un paquete desencapsulado no debe ser reenviado a menos que el nodo haya sido explícitamente configurado para reenviar tales paquetes hacia la dirección IPv4 de origen.

2.7.3.5 Descubrimiento de vecinos sobre túneles

Los túneles automáticos y los configurados unidireccionalmente son considerados para trabajar en una sola dirección. Así el único aspecto del descubrimiento de vecino y la autoconfiguración stateless que se aplica a estos túneles es la formación de la dirección de enlace local.

Si una implementación proporciona una configuración bidireccional de túnel, debe al menos aceptar y responder a los paquetes de prueba usados por la NUD[33]. Tales implementaciones también deberían enviar paquetes de prueba NUD para detectar cuando el túnel configurado falla, y el punto en el cual la implementación puede usar una ruta alternativa para alcanzar el destino. Cabe recalcar que el descubrimiento de vecino permite que el envío de pruebas NUD sea omitido para enlaces router a router, si el protocolo de ruteo detecta disponibilidad bidireccional.

Para los propósitos del descubrimiento de vecino, se asume que tanto los túneles automáticos como los configurados no tienen una dirección de enlace

[33] NUD: Neighbor Unreachability Detection - Detección de Vecino Inalcanzable

local, aunque la capa de enlace (IPv4) no tiene dirección. Esto significa que un remitente de paquetes de descubrimiento de vecino:

- No debe incluir las opciones de la dirección del origen del enlace local o las opciones de la dirección de destino del enlace local, en el enlace del túnel.
- Debe ignorar tácitamente cualquier opción SLLA o TLLA recibida en el enlace del túnel.

2.7.4 Túneles IPv6 configurados manualmente

El uso principal de un túnel configurado es proveer conexiones estables y seguras para la comunicación entre dos routers de borde (edge routers), ó entre un sistema y un router de borde, ó para la conexión a redes remotas IPv6. Los routers de borde y los sistemas usados como puntos finales de los túneles deben ser dispositivos dual stack. Los túneles manuales son usados entre dos puntos y requieren de una configuración en las direcciones de origen y destino del túnel, mientras que los túneles automáticos necesitan solamente ser habilitados y son temporales.

Como en otros mecanismos de tunneling, NAT[34] no es permitida a lo largo del túnel. La siguiente figura muestra la estructura de un túnel configurado manualmente.

[34] NAT: Network Address Translation - Traducción de Direcciones de Red

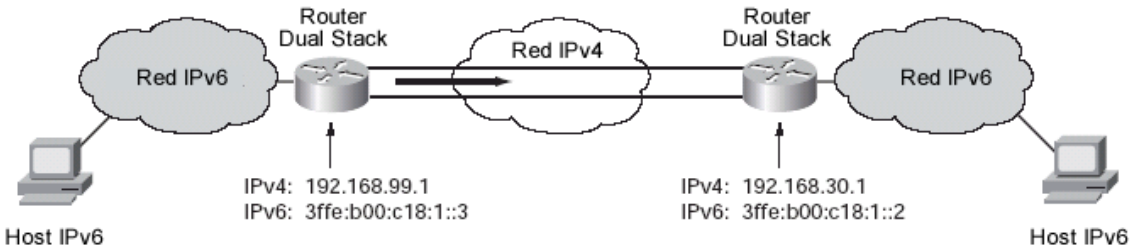


Figura 2.30: Túnel configurado manualmente

2.7.5 IPv6 sobre túneles GRE[35] IPv4

Los túneles IPv6 sobre túneles GRE IPv4 usa la técnica estándar de tunneling GRE, que fue diseñada para proveer los servicios necesarios para implementar cualquier esquema de encapsulación punto-a-punto. Estos túneles son enlaces entre dos puntos, con un túnel separado para cada enlace.

De manera similar que los túneles configurados manualmente, los túneles GRE son usados entre dos puntos que requieren de una configuración en la direcciones de origen y destino. Los routers de borde y los sistemas finales usados como puntos finales de los túneles deben ser dual stack.

Debido a que el protocolo de ruteo IS-IS trabaja en la capa de enlace, otras técnicas de tunneling aparte de GRE no pueden ser utilizadas. La siguiente figura muestra como un paquete IPv6 es enviado a través de un túnel GRE.

[35] GRE: Generic Route Encapsulation

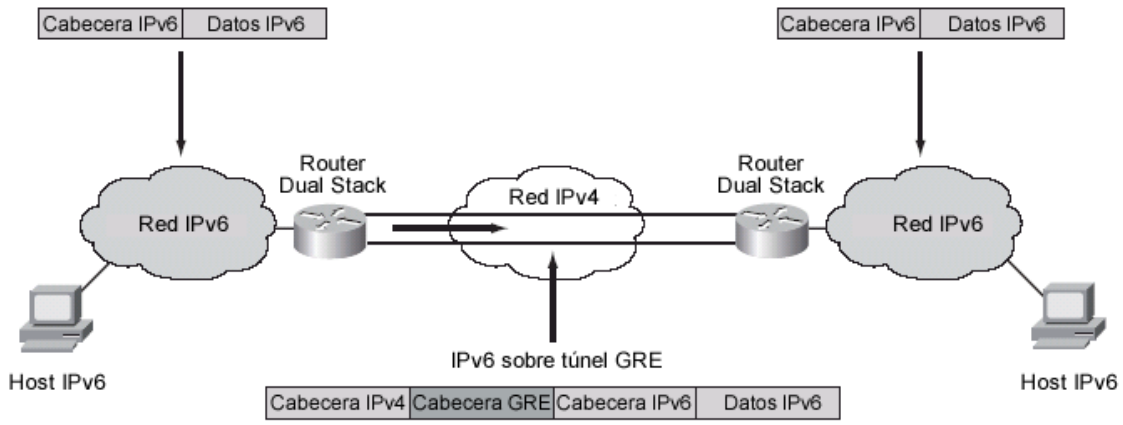


Figura 2.31: IPv6 sobre túneles GRE

Como en el caso de los túneles configurados manualmente, se deben configurar las direcciones IPv4 e IPv6 del router dual stack en la interfase del túnel GRE, e identificar los puntos de entrada y salida del túnel (origen y destino) usando direcciones IPv4.

2.7.6 Túneles automáticos compatibles con IPv4

Un túnel automático compatible con IPv4 es un mecanismo de tunneling IPv6 sobre IPv4, que usa una dirección IPv6 compatible con IPv4. Esta dirección consta de una concatenación de ceros en los primeros 96 bits de la izquierda y una dirección IPv4 embebida en los últimos 32 bits. Por ejemplo ::192.168.99.1 es una dirección IPv6 compatible con IPv4.

A pesar de que pueden ser configurados entre sistemas finales, routers de borde, ó entre ellos, los túneles automáticos compatibles con IPv4 han sido principalmente usados para establecer conexiones entre routers.

A diferencia de los túneles manuales, la técnica de túneles compatibles con IPv4 construye túneles con nodos remotos que pueden cambiar continuamente sus configuraciones. No se requiere de la configuración manual de los puntos finales del túnel ya que la dirección de destino es determinada automáticamente por la dirección IPv4. Los túneles automáticos son inicializados y finalizados de la forma en la que se requiera, y duran tanto como dura la comunicación. A continuación se muestra la configuración de un túnel automático compatible con IPv4.

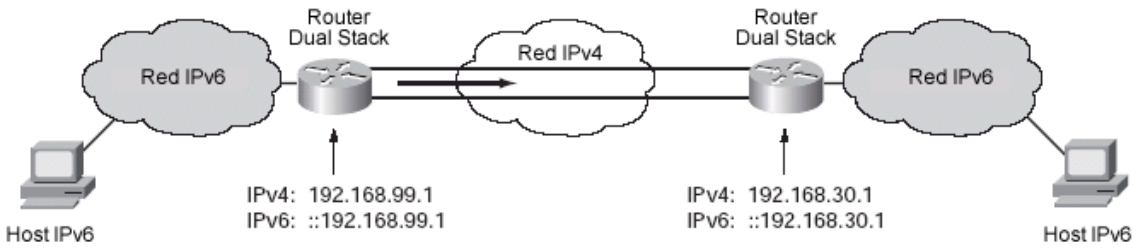


Figura 2.32: Túnel automático compatible con IPv4

A pesar de que esta técnica es una forma fácil de crear túneles, no contribuye de buena manera al desarrollo de IPv6 en redes ya que cada host requiere de una dirección IPv4, lo que va en desmedro del amplio espacio de direccionamiento de IPv6. Actualmente el tunneling compatible con IPv4 ha sido reemplazado por el mecanismo de túneles automáticos 6to4. Así pues, el uso de túneles compatibles con IPv4 como mecanismo de transición será próximamente desaprobado.

2.7.7 Túneles Automáticos 6to4

Un túnel automático 6to4 permite a los dominios IPv6 ser conectados sobre una red IPv4 y permite las conexiones a redes remotas IPv6. El escenario más simple para la implementación de túneles 6to4 es interconectar múltiples sitios, cada uno de los cuales tiene al menos una conexión a una red IPv4. Esta red IPv4 puede ser desde la red global Internet ó el backbone de una red corporativa.

El túnel 6to4 trata a la infraestructura IPv4 como un enlace virtual sin broadcast, usando una dirección IPv4 embebida en una dirección IPv6 para encontrar el otro punto final del túnel. Cada dominio IPv6 requiere de un router dual stack que automáticamente construya el túnel IPv4 usando un único prefijo de ruteo 2002::/16 en la dirección IPv6 con la dirección IPv4 del destino túnel concatenada al prefijo de ruteo antes citado. El requerimiento clave es que cada sitio tenga una dirección IPv6. Cada sitio, aún si tiene una sola dirección pública IPv4, tiene un único prefijo de ruteo en IPv6. La siguiente figura muestra la configuración de un túnel 6to4.

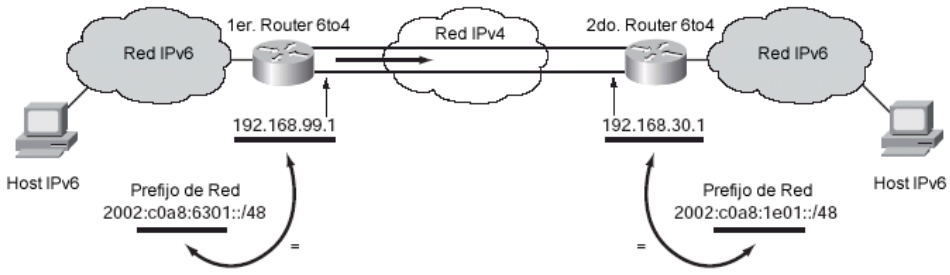


Figura 2.33: Túnel Automático 6to4

Se recomienda que cada sitio tenga solamente una dirección 6to4 asignada a la interfase externa del router. Todos los sitios necesitan ejecutar un protocolo

interior de ruteo IPv6, como por ejemplo el RIPng[36] para el ruteo IPv6 dentro del sitio; el ruteo exterior es manejado por el protocolo de ruteo exterior IPv4.

2.7.8 Esquema de trabajo del Protocolo TSP

El Protocolo de Configuración de Túneles (TSP[37]) es un protocolo de señalización que permite configurar los parámetros de tunneling entre los dos puntos finales de un túnel. Uno de los extremos corresponde al cliente TSP que solicita el establecimiento del túnel, mientras que el otro extremo es el servidor que configurará el servicio de tunneling.

Dentro de una sesión, el protocolo TSP realiza el siguiente procedimiento de comunicación entre los extremos del túnel:

- Autenticación de los usuarios
- Encapsulamiento del túnel (p.ej. IPv6 sobre IPv4)
- Asignación de direcciones IP para los extremos del túnel
- Asignación de prefijos IPv6, cuando el cliente es un router y sirve a una red local
- Resolución DNS en base al prefijo IPv6
- Registro del DNS en el extremo final del túnel
- Protocolos de enrutamiento

[36] RIPng: Router Information Protocol Next Generation – Protocolo de Información de Ruteo de Próxima Generación
[37] TSP: Tunnel Setup Protocol

El encapsulamiento del túnel puede ser especificado explícitamente por parte del cliente, o puede ser determinado durante el intercambio de información del protocolo TSP.

Los nodos que participan en el esquema de trabajo TSP son:

1. Cliente TSP
2. Punto final del cliente del túnel
3. Servidor TSP
4. Punto final del servidor del túnel

Los nodos 1, 3 y 4 forman el Modelo de Tunnel Broker[38], donde el nodo 3 es el Broker y el nodo 4 es el Servidor de Túneles. Tal como se muestra en la siguiente figura, un Tunnel Broker puede controlar uno o más servidores de túneles.

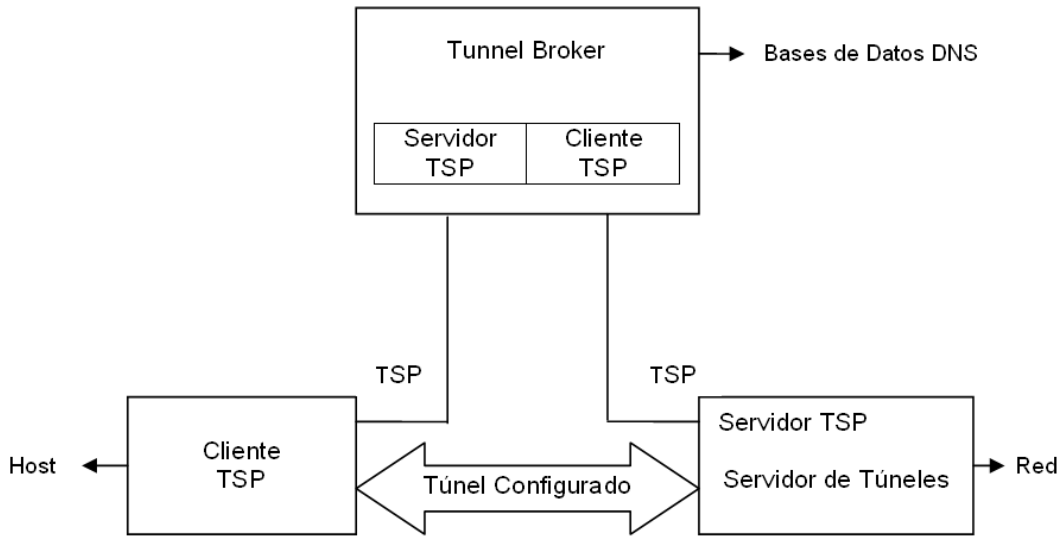


Figura 2.34: Esquema de trabajo de un Tunnel Broker

[38] RFC 3053: IPv6 Tunnel Broker

En su modelo más simple, un nodo es el cliente configurado como un extremo del túnel (el nodo1 y 2 son uno solo), y el segundo es el servidor configurado como el otro extremo del túnel (el nodo 3 y 4 son uno solo).

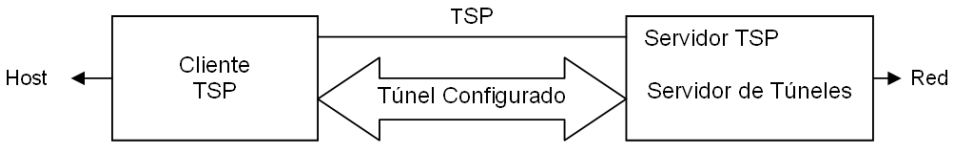


Figura 2.35: Túnel entre un cliente TSP y un servidor de túneles

Desde el punto de vista de un sistema operativo, el protocolo TSP es implementado como una aplicación cliente, la cual es capaz de configurar parámetros de red del sistema operativo.

2.7.8.1 Ventajas del Protocolo TSP

- Permite establecer un túnel configurado
- Posee un método de señalización flexible y extenso (XML, SASL)
- Ofrece una sola solución para varios mecanismos de encapsulación: IPv6 en IPv4, IPv4 en IPv6, IPv6 sobre UDPv4, etc.
- Asignación de prefijos
- Resolución de nombres DNS
- Movilidad del nodo IP subyacente
- Los túneles establecidos por TSP son configurados manualmente, permitiendo mayores niveles de seguridad que los túneles automáticos.

2.7.8.2 Descripción del Protocolo TSP

2.7.8.2.1 Terminología

- Tunnel Broker: En el modelo de trabajo del Tunnel Broker, el broker se hace cargo de todas las comunicaciones entre los servidores de túneles y los clientes de túneles.

  Los clientes buscan un broker para establecer un túnel y el broker se encarga de encontrar el servidor de túneles adecuado para la comunicación, luego el broker pide al servidor de túneles establecer la configuración del túnel para luego enviar la información al cliente del túnel.
- Servidor de Túneles: Los servidores proveen el servicio de túneles a los clientes. Pueden recibir pedidos de túneles desde un broker (como sucede en el modelo de trabajo del Tunnel Broker) o directamente desde el cliente. El servidor de túneles es uno de los puntos finales del túnel.
- Cliente de Túneles: El cliente es la entidad que necesita de un túnel para establecer conectividad entre redes IPv4 e IPv6, a fin de ofrecer o solicitar ciertos servicios. El cliente de túneles es el otro punto final del túnel, y puede ser un host ó un router.

2.7.8.2.2 Descripción General

El TSP es iniciado desde un nodo cliente hacia un tunnel broker, y tiene tres fases:

1. Fase de Autenticación: El broker ó el servidor de túneles informa de su capacidad al cliente, y cuando el cliente se autentica contra el broker ó servidor de túneles.
2. Fase de Comandos: El cliente solicita un nuevo túnel, ó pide actualizar uno existente.
3. Fase de Respuesta: El cliente recibe la respuesta al pedido de túnel desde el broker ó el servidor, y el cliente acepta o rechaza el túnel ofrecido.

Para cada comando enviado desde el cliente del túnel existe una respuesta desde el servidor. Después de que la fase de respuesta se ha completado, el túnel es establecido por el cliente.

CAPITULO III

3 CONFIGURACIÓN DE IPV6

3.1 Configuración de IPv6 en Windows

El soporte para IPv6 en los sistemas operativos de Microsoft está dado a partir de la versión Windows 2000 Server. Para las versiones 2000 Server y XP es necesario cumplir con ciertos requisitos a fin de lograr una total funcionalidad, mientras que para Windows Server 2003 el soporte para IPv6 es una característica incorporada por defecto.

Los requisitos que deben instalarse en Windows 2000 Server son los siguientes:

- Service Pack #1 ó superior
- Paquete de soporte IPv6 para Microsoft Windows 2000 Server[39]

Para el sistema operativo Microsoft Windows XP es necesaria la instalación del Service Pack #1 ó superior.

[39] Descargar del sitio: http://msdn.microsoft.com/downloads/sdks/platform/tpipv6/ReadMe.asp

3.1.1 Instalación en Windows 2000 Server

1. Autenticarse en el equipo Windows 2000 Server con una cuenta con privilegios de administrador.
2. Usando el Explorador de Windows, ejecutar el archivo *setup.exe* desde la ubicación en la cual se descomprimieron los archivos del paquete de soporte IPv6; por ejemplo D:\IPv6TP.

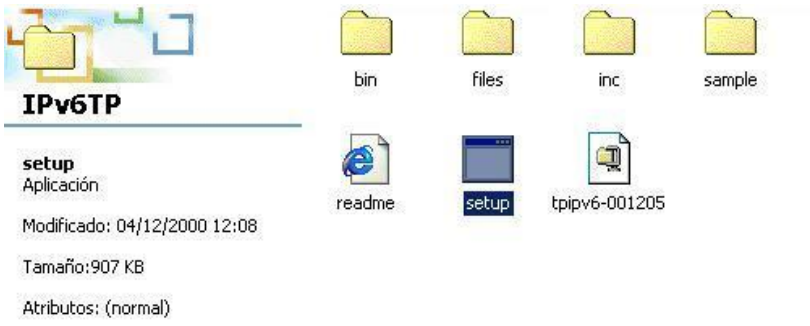


Figura 3.1: Instalación del protocolo IPv6 en Windows 2000 Server

3. Abrir la ventana de Conexiones de Red y Acceso Telefónico desde Inicio / Configuración / Panel de Control ó en la opción *Propiedades* del menú secundario del icono de Mis sitios de Red en el escritorio de Windows.

Figura 3.2: Acceso a las propiedades de Mis sitios de Red

4. Dar clic derecho sobre el icono correspondiente a la conexión de red local y luego en la opción *Propiedades*.

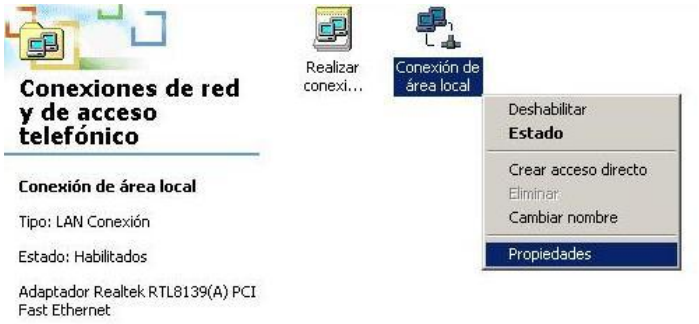


Figura 3.3: Propiedades de la conexión de área local

5. Pulsar el botón *Instalar*

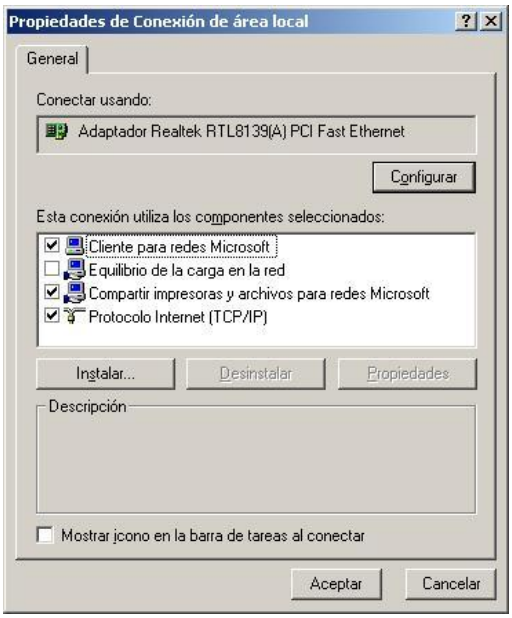


Figura 3.4: Ventana de propiedades de Conexión de área local

6. En la ventana para *Seleccionar tipo de componente de red* escoger la opción *Protocolo* y luego dar clic en *Agregar*.

Figura 3.5: Ventana de selección de componente de red

7. Luego, en la ventana *Seleccione el protocolo de red* elegir el protocolo Microsoft IPv6 y dar clic en Aceptar

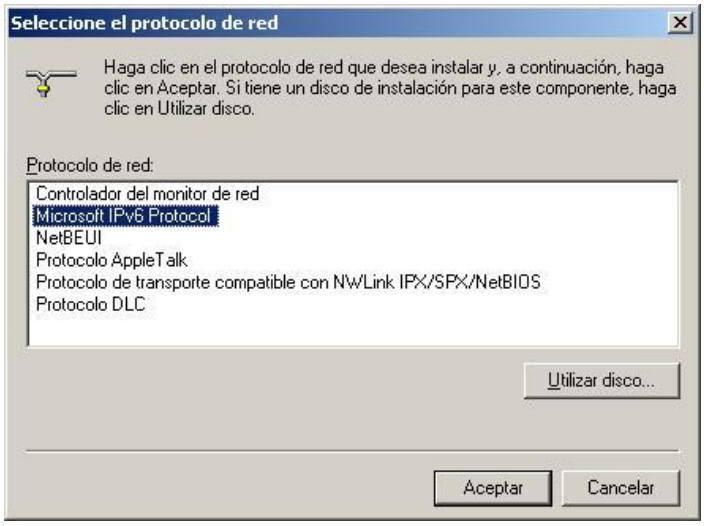


Figura 3.6: Ventana de selección de protocolo de red

8. Cerrar la ventana de Propiedades de Conexión de Área Local

Una vez finalizado este proceso, el protocolo Microsoft IPv6 es incorporado a todas las interfases de red del equipo. Para poder visualizar la configuración de las mismas se puede utilizar el comando *ipv6 if*, como se detalla a continuación:

Figura 3.7: Descripción de la interfaz de red

Como se puede observar, la aplicación de soporte para IPv6 asigna por defecto una dirección de enlace local para cada interfase de red del equipo, en base a la dirección MAC de la misma, y se asignan otras direcciones como la de multicast y de loopback. Cabe recalcar que si en la red donde se encuentra el equipo existiese un router, se asignan otras direcciones en base a la información enviada por el mismo.

Otros comandos útiles como ping6 y tracert6, tienen la misma funcionalidad que sus similares en IPv4, y también están listos para ser utilizados cuando sean requeridos. A continuación se muestran ejemplos utilizando el comando ping6 hacia la dirección de loopback (::1) y de enlace local:

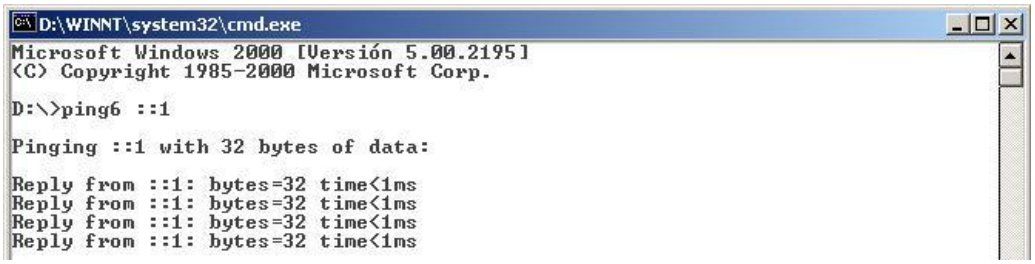


Figura 3.8: Ping a la dirección de loop back

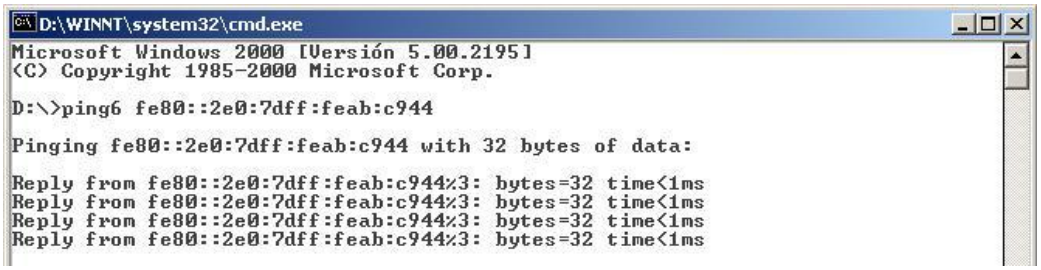


Figura 3.9: Ping a la dirección de enlace local

3.1.2 Instalación en Windows XP

El proceso para incorporar el protocolo IPv6 a Windows XP es relativamente sencillo, y solamente requiere que previamente se haya instalado el Service Pack #1. Esto se puede verificar en la ventana de Propiedades del sistema, a la cual se accede dando clic derecho sobre el icono de Mi PC en el escritorio de Windows XP, o mediante el icono de Sistema en el Panel de Control.

Figura 3.10: Ventana de propiedades del sistema

Si después de realizar esta verificación se determina que el Service Pack #1 no está instalado, es necesario conseguir un medio de instalación a fin de incorporar la utilidad del paquete al sistema operativo que necesita el soporte para IPv6. Una vez que se tenga la total seguridad de que el Service Pack #1 está instalado, se puede continuar con el proceso de habilitación del protocolo IPv6; para ello se debe abrir la ventana de Conexiones de Red y Acceso Telefónico, sea desde el Panel de Control o dando clic derecho sobre el icono de Mis sitios de red.

Figura 3.11: Propiedades de Mis sitios de red

A continuación se debe acceder a las Propiedades de Conexión de área local por medio de un clic derecho sobre el icono de la interfase de red en la cual se desea agregar el protocolo IPv6.

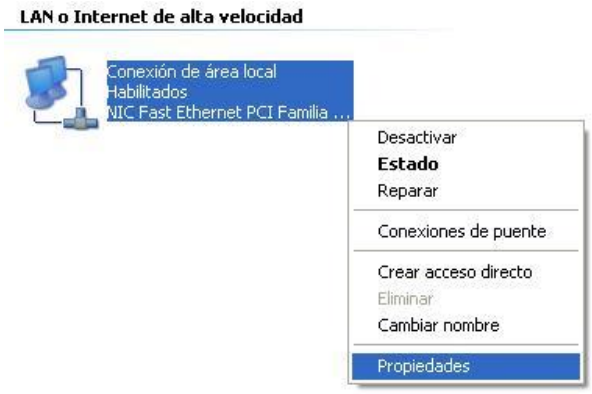


Figura 3.12: Propiedades de la Conexión de área local

En la ventana que aparece a continuación se debe elegir la opcion Instalar.

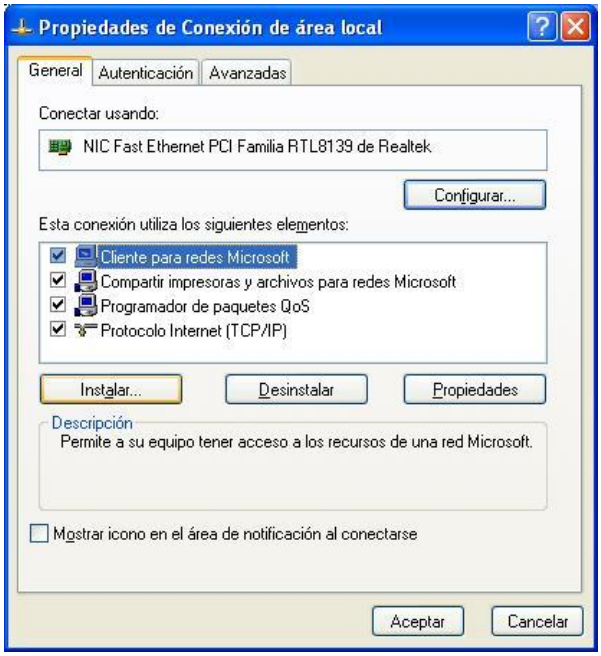


Figura 3.13: Ventana de las propiedades de conexión de área local

Luego de esto el sistema brinda la posibilidad de agregar un cliente, un servicio o un protocolo. Se procede a elegir la instalación de un nuevo protocolo.

Figura 3.14: Ventana de selección de componente de red

Luego se muestra una lista de protocolos de entre los cuales se debe seleccionar el requerido para el caso de estudio.

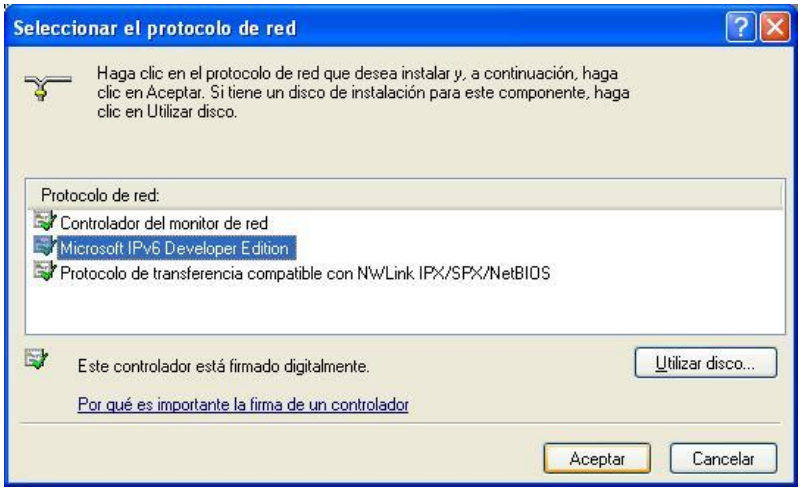


Figura 3.15: Ventana de selección del protocolo de red

Por último y para finalizar el proceso de instalación se deben aceptar los cambios realizados y cerrar la ventana de propiedades de la conexión de área local.

Figura 3.16: Descripción de la interfaz de red

A fin de comprobar la funcionalidad del protocolo instalado se pueden hacer varias pruebas, como por ejemplo las del comando ping hacia la dirección de loopback (::1) y la dirección de enlace local asignada automáticamente en base a la dirección MAC de la interfase de red.

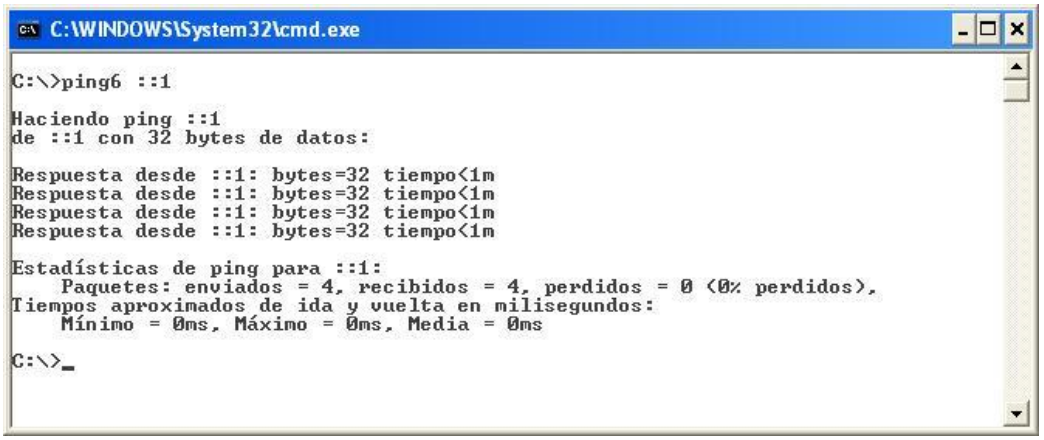


Figura 3.17: Ping a la dirección de loopback

Figura 3.18: Ping a la dirección de enlace local

3.1.3 Instalación en Windows Server 2003

El procedimiento a seguir para instalar el protocolo IPv6 en Windows Server 2003 es similar al tratado en Windows XP, con la única diferencia de que no es necesaria la verificación previa de la instalación del Service Pack #1, pues en la versión Server 2003 el soporte para IPv6 es una característica incorporada por defecto.

3.2 Configuración IPv6 en LINUX

3.2.1 Introducción

El protocolo IPv6 ofrece varias ventajas como se ha mencionado anteriormente y se pretende mostrar su funcionamiento mediante la implementación de dicho protocolo sobre una plataforma Linux.

El protocolo IPv6 se implementa como un módulo y para que no exista ningún inconveniente con su funcionalidad es necesario que el sistema operativo cumpla con un requisito esencial, que cuente con una versión 2.2.x ó superior de Kernel[40], en las cuales el soporte del módulo IPv6 viene incorporado. Sin embargo se debe verificar que dicho módulo se cargue al arrancar el sistema operativo.

La versión del sistema operativo con el que se va a trabajar para la implementación del caso de estudio es Linux Red Hat 8.0 y Fedora. Estos sistemas operativos cumplen con los requisitos de sistema necesarios para trabajar con IPv6.

3.2.2 Instalación de IPv6

El proceso de instalación del protocolo IPv6 comienza con la ejecución de los siguientes comandos, que permiten la habilitación del módulo IPv6 para Linux.

[40] Kernel: También conocido como núcleo es aquella parte del sistema operativo que interactúa de forma directa con el hardware de una máquina.

```
[root@ netxpertsconsulting root]#cd /
[root@ netxpertsconsulting /]#insmod ipv6
Using /lib/modules/2.4.20-8/kernel/net/ipv6/ipv6.o
```

Luego se deben editar manualmente dos archivos de configuración, a fin de que el protocolo IPv6 quede incorporado a la funcionalidad del sistema operativo. El primer archivo se denomina **ifcfg-eth0** y especifica el nombre del dispositivo de la interfaz de red, la dirección IP, la máscara de red, la dirección de red, la dirección de broadcast e información sobre el modo de arranque. En este archivo se debe aumentar una línea que permitirá la funcionalidad del protocolo IPv6. Utilizando cualquier editor de texto y utilizando una ventana de comandos se edita el archivo ifcfg-eth0 como se indica a continuación:

```
vi etc/sysconfig/network-scripts/ifcfg-eth0
IPV6INIT= yes
```

El segundo archivo se denomina **network**, y contiene información relevante acerca del entorno de red del equipo. De la misma forma que en el caso anterior, se aumenta una línea como se muestra a continuación:

```
vi etc/sysconfig/network
NETWORKING_IPV6= yes
```

Cabe recalcar que si no se editan los dos archivos referidos luego de instalar el módulo IPv6, los cambios no serán permanentes y se perderán al arrancar nuevamente el sistema operativo.

3.2.3 Prueba de funcionalidad

Para probar la funcionalidad del protocolo IPv6 sobre la plataforma Linux, se escribe el comando **ifconfig** en una ventana de comandos, el cual desplegará toda la información de configuración de la red como se muestra a continuación:

```
[root@ netxpertsconsulting /]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:07:95:DC:CA:C5
          inet addr:10.10.0.50  Bcast:10.10.0.61  Mask:255.255.255.240
          inet6 addr: fe80::207:95ff:fedc:cac5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:182 errors:0 dropped:0 overruns:0 frame:0
          TX packets:56 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:19703 (19.2 Kb)  TX bytes:4363 (4.2 Kb)
          Interrupt:11 Base address:0xcc00

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:5759 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5759 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:392833 (383.6 Kb)  TX bytes:392833 (383.6 Kb)
```

Como se puede observar en la interfase **eth0** se creó la dirección de link: **fe80::207:95ff:fedc:cac5/64,** con lo que queda comprobado que el protocolo IPv6 se habilitó normalmente. Se pueden realizar pruebas de **ping6** a la dirección de loopback y de link para comprobar la instalación del protocolo.

```
[root@netxpertsconsulting /]# ping6 ::1
PING ::1(::1) 56 data bytes
64 bytes from ::1: icmp_seq=0 ttl=64 time=0.066 ms
64 bytes from ::1: icmp_seq=1 ttl=64 time=0.056 ms
64 bytes from ::1: icmp_seq=2 ttl=64 time=0.050 ms
64 bytes from ::1: icmp_seq=3 ttl=64 time=0.057 ms

--- ::1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3018ms
rtt min/avg/max/mdev = 0.050/0.057/0.066/0.007 ms, pipe 2
```

```
[root@netxpertsconsulting /]# ping6 fe80::250:4ff:fea5:6b94 -I eth0
PING fe80::250:4ff:fea5:6b94(fe80::250:4ff:fea5:6b94) from ::1 eth0:
56 data bytes
64 bytes from fe80::250:4ff:fea5:6b94: icmp_seq=0 ttl=64 time=0.081 ms
64 bytes from fe80::250:4ff:fea5:6b94: icmp_seq=1 ttl=64 time=0.086 ms
```

```
64 bytes from fe80::250:4ff:fea5:6b94: icmp_seq=2 ttl=64 time=0.075 ms
64 bytes from fe80::250:4ff:fea5:6b94: icmp_seq=3 ttl=64 time=0.078 ms

--- fe80::250:4ff:fea5:6b94 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3014ms
rtt min/avg/max/mdev = 0.075/0.080/0.086/0.004 ms, pipe 2
```

3.3 Configuración de IPv6 en Routers

3.3.1 Introducción

Los routers necesitan de sistemas operativos para ejecutar aplicaciones de software, de igual forma como sucede en los computadores. Por medio de estos sistemas operativos se pueden ejecutar diversos archivos de configuración, los cuales controlan el flujo de tráfico del router, en base al uso de los protocolos y tablas de ruteo. Es necesario cerciorarse de que la versión del sistema operativo instalada en los routers incluya el soporte para el protocolo IPv6, a fin de que se pueda realizar sin problemas la implementación de los túneles.

Para el caso de estudio se utilizaron routers Cisco, los cuales se componen básicamente de las siguientes partes:

- RAM/DRAM: Almacena tablas de enrutamiento, caché ARP, caché de conmutación rápida, buffering de paquetes (RAM compartida) y colas de espera de paquetes. La RAM también proporciona memoria temporal y/o de ejecución para el archivo de configuración del router, mientras está encendido; ya que cuando se apaga o reinicia, el contenido de la RAM se pierde.

- NVRAM: RAM no volátil. Almacena el archivo de configuración de inicio/copia de respaldo del router. El contenido no se elimina cuando se apaga o reinicia el router.
- Flash: ROM borrable y reprogramable. Contiene la imagen y microcódigo del sistema operativo. Permite actualizar el software sin eliminar y reemplazar chips en el procesador; el contenido se conserva cuando se apaga o reinicia el router. Se pueden almacenar múltiples versiones del software IOS en la memoria Flash
- ROM: Contiene diagnósticos de encendido, un programa bootstrap y el software del sistema operativo. Las actualizaciones de software en la ROM requieren el reemplazo físico de partes en el equipo.
- Interfaz: Conexión de red a través de la cual los paquetes entran y salen de un router.

Los routers se pueden usar para segmentar dispositivos de redes LAN, pero su uso principal es en las redes WAN. Las dos funciones principales de los routers son la selección de mejores rutas para los paquetes de datos entrantes, y la conmutación de paquetes a la interfaz de salida correspondiente. Los routers hacen esto creando tablas de enrutamiento e intercambiando la información de red de dichas tablas con otros routers.

3.3.2 Configuración de Routers CISCO

Para configurar los routers Cisco, se debe acceder a la interfaz de usuario en el router con un terminal o acceder de forma remota. Por razones de seguridad el router tiene dos niveles de acceso:

- Modo usuario: Permite efectuar una verificación del estado del router, más no acceder a la configuración.
- Modo privilegiado: Permite realizar cambios en la configuración del router.

3.3.3 Ingresar en modo privilegiado

Para acceder al conjunto completo de comandos, se debe habilitar el modo privilegiado. En el indicador "mayor que" **(>)** se debe escribir **enable** o **ena** (forma abreviada), para luego escribir la contraseña del router. Si los datos de autenticación son correctos el indicador se transforma en un signo de número **(#)**, lo que indica que se puede trabajar en el modo privilegiado.

Comando	Comentario
Router>**enable** Password>******* Router**#**	*Modo privilegiado*

Para salir de la configuración del router es necesario que primero se deshabilite el modo privilegiado, para luego usar el comando **exit** (salir).

Comando	Comentario

Router#**disable** Router**>exit**	*Deshabilitar modo privilegiado*

3.3.4 Configuración del nombre del equipo

Se utiliza el comando **configure terminal** para ingresar al modo de configuración de terminal. Una vez dentro se ingresa el comando **hostname x**, donde x representa el nombre que se quiere dar al dispositivo. Para salir del modo de configuración de terminal se utiliza las teclas **Ctrl + Z.**

Para el caso de estudio se utilizaron 3 routers, siendo el proceso de asignación de nombres similar para todos. La nomenclatura utilizada es la siguiente:

- REDFISI: 1ra. red IPv6
- INTERNET: Simula la Internet IPv4
- REDESPE: 2da. red IPv6

Router#**config terminal**	*Ingreso al modo de configuración*
Enter configuration commands, one per line. End with CNTL/Z.	
Router(config)#**hostname x**	*Este comando no admite la inclusión de espacios dentro del nombre del dispositivo.*

3.3.5 Configuración de los puertos

Para configurar los puertos se debe ingresar el comando: **interface xy**, donde **x** es el nombre del tipo de interfaz e **y** corresponde al número que identifica dicha interfaz. (p. ej. interface serial0). Para los routers dual stack del caso de

estudio, se configuraron las interfaces ethernet0 y serial0, mientras que para el router IPv4 se configuraron las interfaces serial0 y serial1.

Para asignar una dirección IPv4 a una interface, se utiliza el comando: **ip address w z**, donde **w** representa la dirección IPv4 y **z** es el valor de la máscara de subred. Para el caso de IPv6 se usa el comando: **ipv6 address q**, donde **q** representa la dirección IPv6 de la interface.

3.3.6 Configuración de rutas estáticas

Una ruta estática es una vía que el router utiliza para transmitir un flujo de tráfico por medio de una interfaz predefinida. En el caso de estudio se utilizan las rutas estáticas para enviar el tráfico IPv4 por las interfaces seriales, mientras que el tráfico IPv6 se envía por el túnel.

Para configurar una ruta estática en IPv4 se usa el comando: **ip route x y z**, donde **x** representa la dirección IP de la red de origen, **y** representa la máscara de la red de origen y **z** representa la dirección IP de la red de destino. Para configurar una ruta estática en IPv6 se usa el comando: **ipv6 route x y**, donde **x** representa la dirección IPv6 de la red de origen e **y** representa la interfaz por donde se enviará el tráfico.

3.3.7 Parámetros generales

La seguridad es un aspecto a tomar en cuenta en los ámbitos de comunicaciones, por lo cual es necesario que se establezca una contraseña para el acceso a las configuraciones de los routers. El comando **enable password x** permite establecer una clave de acceso al modo privilegiado, lo cual evitará que terceras personas puedan manipular las configuraciones de los equipos, donde **x** es la contraseña establecida por el administrador del router. Además es recomendable utilizar la encriptación de los passwords, para lo cual se utiliza el comando **enable secret class**.

```
Router(config)#enable password *****
                                    Establece el password para el modo
                                    privilegiado.
Router(config)#enable secret class
                                    Habilita la encriptación para el
                                    password.
```

Otro aspecto referente a la seguridad se refiere al bloqueo para impedir que se realicen configuraciones utilizando un cliente web. Para ello se utiliza el comando **no ip http server**.

```
Router(config)#no ip http server
                                    No permite configuraciones
                                    utilizando un cliente web.
```

Otro aspecto tomado en cuenta en el caso de estudio fue la habilitación de la subred cero, para lo cual se utiliza el comando **ip subnet-zero**.

```
Router(config)#ip subnet-zero
                                    Al realizar subredes, Cisco
                                    considera que la dirección de red
                                    de la clase no puede ser utilizada
                                    como subred y a través de este
                                    comando habilitamos el que se pueda
                                    trabajar con la subred cero
```

Luego de configurar todos los aspectos relacionados a la configuración de los routers, es necesario guardar los cambios realizados y asegurar que los mismos permanezcan luego de que el dispositivo se apague o reinicie.

```
                              Escribe los cambios sobre la memoria del
Router#wri mem                router para que no se eliminen al apagar o
                              reiniciar.
Building configuration…….
[OK]
```

CAPITULO IV

4 IMPLEMENTACIÓN DE TUNNELING

4.1 Introducción

Se denomina Tunneling a un mecanismo que permite encapsular tráfico IPv6 en paquetes IPv4, a fin de que el mismo pueda utilizar la infraestructura de redes existente en la actualidad, sin necesidad de incurrir en un cambio de tecnología para permitir la coexistencia entre redes IPv4 e IPv6. Debido a las grandes ventajas que ofrece, el tunneling es una de las alternativas más utilizadas en redes con soporte para ambos protocolos.

Gracias a la utilización del tunneling es posible ofrecer servicios IPv6 end-to-end sin causar un impacto negativo en el rendimiento de los sistemas IPv4 por los cuales viaja el tráfico IPv6 encapsulado. Las alternativas son varias, conexión entre redes IPv6, desde redes IPv4 a IPv6 y viceversa, y, la conexión a redes remotas IPv6, tales como el 6bone.

Existen varios mecanismos de tunneling, que básicamente se dividen en dos categorías: túneles configurados manualmente y túneles automáticos. Todos los mecanismos comparten un requerimiento básico, el soporte dual stack en los extremos del túnel, es decir, en el origen y el destino del túnel debe existir equipos capaces de trabajar tanto con IPv4 como con IPv6.

4.2 Conexión entre redes IPv6 sobre infraestructura de comunicaciones IPv4

4.2.1 Finalidad de la práctica

4.2.1.1 Realizar la configuración de los routers a fin de establecer un túnel manual para conectar dos redes IPv6.

4.2.1.2 Realizar la configuración de los routers a fin de establecer un túnel GRE para conectar dos redes IPv6.

4.2.1.3 Realizar la configuración de los routers a fin de establecer un túnel automático para conectar dos redes IPv6.

4.2.2 Equipos utilizados

4.2.2.1 2 routers Cisco con soporte dual stack

4.2.2.2 1 router Cisco con soporte para IPv4

4.2.2.3 1 PC con sistema operativo Windows

4.2.2.4 1 PC con sistema operativo Linux

4.2.3 Descripción de la red

A fin de implementar los diversos mecanismos de tunneling se utilizó una infraestructura de red que simula la interconexión de dos redes IPv6 conectadas a través de una red IPv4 existente como el Internet.

La primera red IPv6 cuenta con un router CISCO modelo 1700 (IOS 12.3), al cual se le instaló una versión actualizada del IOS que tiene soporte para el protocolo IPv6, con lo que se convierte en un router dual stack. Este router, denominado “REDFISI”, tiene un puerto serial y un puerto Ethernet, el primero de los cuales se utiliza para conectarse al router de la red IPv4, mientras que el segundo es utilizado para conectar un concentrador al que se unirán los hosts de la red IPv6. Para el caso de estudio se utilizó un computador con el sistema operativo Fedora2 de la plataforma Linux.

La segunda red IPv6 posee un router dual stack CISCO modelo 2500 (IOS 12.2), denominado “REDESPE”, el cual posee 2 puertos seriales y 16 puertos Ethernet. Uno de los puertos seriales es utilizado para la conexión a la red IPv4, mientras que los puertos ethernet se constituyen en un hub al cual se pueden conectar directamente los hosts de la red IPv6. Para el caso de estudio se utilizó un computador con el sistema operativo Windows XP de la plataforma Windows.

A efectos de simular la infraestructura de las redes IPv4 actuales se utilizó un router CISCO 2500 con soporte único de IPv4, denominado “INTERNET”, el

mismo que posee dos interfases seriales utilizadas para la conexión con los routers dual stack de las redes IPv6.

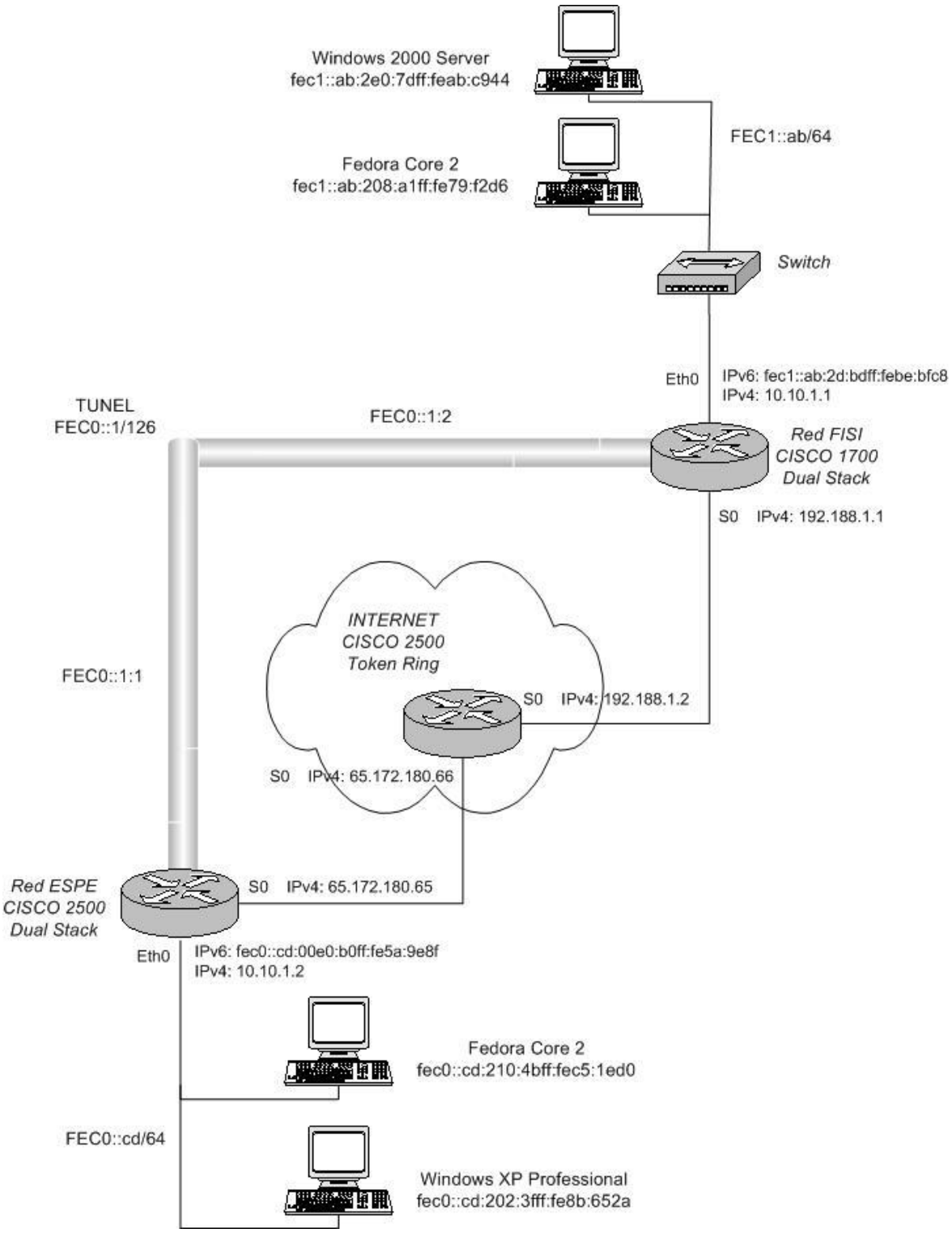


Figura 4.1: Descripción de la red

4.2.4 Implementación de Tunneling Manual

4.2.4.1 Router: REDFISI

```
Router#config terminal

                                    Configuración de terminal

Router(config)#hostname REDFISI

                                    Nombre del Router
```

4.2.4.2 Configuración de interfases

```
REDFISI(config)#interface FastEthernet0

                                    Configuración de la interfase Ethernet
                                    que tendrá direcciones IPv4 e IPv6

REDFISI(config-if)#ip address 10.10.1.1 255.255.255.0

                                    Asigna la dirección IPv4 y la máscara
                                    de subred

REDFISI(config-if)#ipv6 enable

                                    Habilita IPv6

REDFISI(config-if)#ipv6 address FEC1::AB:0010:7BFF:FEE8:FEE8:120E/64

                                    Asigna la dirección IPv6

REDFISI(config-if)#no shutdown

                                    Habilita administrativamente el puerto

REDFISI(config-if)#exit

REDFISI(config)#interface Serial 0

                                    Configuración de la interfaz Serial

REDFISI(config-if)#ip address 192.188.1.1 255.255.255.252

                                    Asigna la dirección IPv4
```

```
REDFISI(config-if)#no shutdown

                                    Habilita administrativamente el Puerto

REDFISI(config-if)#exit

REDFISI(config)#interface Tunnel 0

                                    Configuración de la interfaz del túnel

REDFISI(config-if)#no ip address

                                    Deshabilita IPv4

REDFISI(config-if)#ipv6 enable

                                    Habilita IPv6

REDFISI(config-if)#ipv6 address FEC0::1:2/126

                                    Asigna la dirección IPv6 al túnel

REDFISI(config-if)#tunnel source Serial0

                                    Establece el origen del túnel

REDFISI(config-if)#tunnel destination 65.172.180.65

                                    Establece el destino del túnel

REDFISI(config-if)#tunnel mode ipv6ip

                                    Establece el modo del túnel manual

REDFISI(config-if)#no shutdown

                                    Habilita administrativamente el puerto

REDFISI(config-if)#exit
```

4.2.4.3 Configuración de rutas estáticas

```
REDFISI(config)#ipv6 unicast-routing

                                    Habilita el ruteo IPv6
```

```
REDFISI(config)#ip route 65.172.180.64 255.255.255.252 192.188.1.2

                                Envía todo el tráfico IPv4 hacia el
                                router INTERNET

REDFISI(config)#ipv6 route ::/0 Tunnel0

                                Envía todo el tráfico IPv6 por la
                                interfaz del túnel
```

4.2.4.4 Router: REDESPE

```
Router#config terminal

                                Configuración de terminal

Router(config)#hostname REDESPE

                                Nombre del Router
```

4.2.4.5 Configuración de interfases

```
REDESPE(config)#interface Ethernet0

                                Configuración de la interfaz Ethernet
                                que poseerá direcciones IPv4 e IPv6

REDESPE(config-if)#ip address 10.10.1.2 255.255.255.0

                                Asigna la dirección IPv4 y la máscara
                                de subred

REDESPE(config-if)#ipv6 enable

                                Habilita IPv6

REDESPE(config-if)#ipv6 address FEC0::CD:0050:54FF:FE80:3D30/64

                                Asigna la dirección IPv6

REDESPE(config-if)#no shutdown

                                Habilita administrativamente el puerto

REDESPE(config-if)#exit
```

```
REDESPE(config)#interface Serial 0
                                    Configuración de la interfaz Serial
REDESPE(config-if)#ip address 65.172.180.65 255.255.255.252
                                    Asigna la dirección IPv4
REDESPE(config-if)#clock rate 2000000
                                    Establece la velocidad de reloj del
                                    puerto, ya que este actúa como DCE.
REDESPE(config-if)#no shutdown
                                    Habilita administrativamente el puerto
REDESPE(config-if)#exit
REDESPE(config)#interface Tunnel 0
                                    Configuración de la interfaz del túnel
REDESPE(config-if)#no ip address
                                    Deshabilita IPv4
REDESPE(config-if)#ipv6 enable
                                    Habilita IPv6
REDESPE(config-if)#ipv6 address FEC0::1:1/126
                                    Asigna la dirección IPv6 al túnel
REDESPE(config-if)#tunnel source Serial0
                                    Establece el origen del túnel
REDESPE(config-if)#tunnel destination 192.188.1.1
                                    Establece el destino del túnel
REDESPE(config-if)#tunnel mode ipv6ip
                                    Establece el modo del túnel manual
```

```
REDESPE(config-if)#no shutdown

                                    Habilita administrativamente el Puerto

REDESPE(config-if)#exit
```

4.2.4.6 Configuración de rutas estáticas

```
REDESPE(config)#ipv6 unicast-routing

                                    Habilita el ruteo IPv6

REDESPE(config)#ip route 192.188.1.0 255.255.255.252 65.172.180.66

                                    Envía todo el tráfico IPv4 hacia el
                                    router INTERNET

REDESPE(config)#ipv6 route ::/0 Tunnel0

                                    Envía todo el tráfico IPv6 por la
                                    interfaz del túnel
```

4.2.4.7 Router: INTERNET

```
Router#config terminal

                                    Configuración de Terminal

Router(config)#hostname INTERNET

                                    Nombre del Router
```

4.2.4.8 Configuración de interfases

```
INTERNET(config)#interface Serial0

                                    Configuración de la interfaz Serial0

INTERNET(config-if)#ip address 65.172.180.66 255.255.255.0

                                    Asigna la dirección IPv4 y la máscara
                                    de subred
```

```
INTERNET(config-if)#no shutdown

                                    Habilita administrativamente el puerto

INTERNET(config-if)#exit

INTERNET(config)#interface Serial1

                                    Configuración de la interfaz Serial1

INTERNET(config-if)#ip address 192.188.1.2 255.255.255.252

                                    Asigna la dirección IPv4 y la máscara
                                    de subred

INTERNET(config-if)#no shutdown

                                    Habilita administrativamente el puerto

INTERNET(config-if)#Ctrl + Z
```

4.2.5 Configuración de Tunneling GRE

Este tipo de túnel es similar al configurado manualmente, con la diferencia de que se usa el modo GRE propietario de CISCO, diseñado para proveer los servicios necesarios para implementar cualquier esquema de encapsulación punto a punto.

Para realizar la configuración de este tipo de túnel se siguen los mismos pasos del mecanismo anterior pero se cambia el modo del túnel, es decir, en lugar de utilizar el modo ipv6ip se usa el modo **gre ip**.

4.2.5.1 Router: REDFISI

```
REDFISI#configure teminal
```

```
REDFISI(config)#interface Serial 0

                                Configuración de la interfaz serial 0

REDFISI(config-if)# encapsulation ppp

                                Define el modo de encapsulamiento de
                                los paquetes

REDFISI(config-if)#exit

REDFISI(config)#interface Tunnel 0

                                Configuración de la interfaz del túnel

REDFISI(config-if)#no ip address

                                Deshabilita IPv4

REDFISI(config-if)#ipv6 enable

                                Habilita IPv6

REDFISI(config-if)#ipv6 address FEC0::1:2/126

                                Asigna la dirección IPv6 al túnel

REDFISI(config-if)#tunnel source Serial0

                                Establece el origen del túnel

REDFISI(config-if)#tunnel destination 65.172.180.65

                                Establece el destino del túnel

REDFISI(config-if)#tunnel mode gre ip

                                Establece el modo del túnel GRE

REDFISI(config-if)#no shutdown

                                Habilita administrativamente el Puerto

REDFISI(config-if)#exit
```

4.2.5.2 Router: REDESPE

```
REDESPE#configure teminal

REDESPE(config)#interface Serial 0

                                Configuración de la interfaz serial 0

REDESPE(config-if)#encapsulation ppp

                                Define el modo de encapsulamiento de
                                los paquetes

REDESPE(config-if)#exit

REDESPE(config)#interface Tunnel 0

                                Configuración de la interfaz del túnel

REDESPE(config-if)#no ip address

                                Deshabilita IPv4

REDESPE(config-if)#ipv6 enable

                                Habilita IPv6

REDESPE(config-if)#ipv6 address FEC0::1:1/126

                                Asigna la dirección IPv6 al túnel

REDESPE(config-if)#tunnel source Serial0

                                Establece el origen del túnel

REDESPE(config-if)#tunnel destination 192.188.1.1

                                Establece el destino del túnel

REDESPE(config-if)#tunnel mode gre ip

                                Establece el modo del túnel GRE

REDESPE(config-if)#no shutdown

                                Habilita administrativamente el Puerto
```

```
REDESPE(config-if)#exit
```

4.2.6 Configuración de Tunneling Automático

4.2.6.1 Router: REDFISI

```
Router#config terminal

                                    Configuración de terminal

Router(config)#hostname REDFISI

                                    Nombre del Router
```

4.2.6.2 Configuración de interfases

```
REDFISI(config)#interface FastEthernet0

                                    Configuración de la interfase Ethernet
                                    que tendrá direcciones IPv4 e IPv6

REDFISI(config-if)#ip address 10.10.1.1 255.255.255.0

                                    Asigna la dirección IPv4 y la máscara
                                    de subred

REDFISI(config-if)#ipv6 enable

                                    Habilita IPv6

REDFISI(config-if)#ipv6 address FEC1::AB:0010:7BFF:FEE8:FEE8:120E/64

                                    Asigna la dirección IPv6

REDFISI(config-if)#exit

REDFISI(config)#interface Serial 0

                                    Configuración de la interfaz Serial

REDFISI(config-if)#ip address 192.188.1.1 255.255.255.252
```

```
                                Asigna la dirección IPv4

REDFISI(config-if)#exit

REDFISI(config)#interface Tunnel 0

                                Configuración de la interfaz del túnel

REDFISI(config-if)#no ip address

                                Deshabilita IPv4

REDFISI(config-if)#ipv6 enable

                                Habilita IPv6

REDFISI(config-if)#ipv6 rip tesis enable

                                Habilita el protocolo de enrutamiento
                                RIP denominado tesis

REDFISI(config-if)#ipv6 address FEC0::1:2/126

                                Asigna la dirección IPv6 al túnel

REDFISI(config-if)#tunnel source Serial0

                                Establece el origen del túnel

REDFISI(config-if)#tunnel mode ipv6ip auto-tunnel

                                Establece el modo del túnel automático

REDFISI(config-if)#no shutdown

                                Habilita administrativamente el Puerto

REDFISI(config-if)#exit
```

4.2.6.3 Configuración de rutas estáticas

```
REDFISI(config)#ipv6 unicast-routing

                                Habilita el ruteo IPv6
```

```
REDFISI(config)#ip route 65.172.180.64 255.255.255.252 192.188.1.2

                                Envía todo el tráfico IPv4 hacia el
                                router INTERNET

REDFISI(config)#ipv6 route ::/0 Tunnel0

                                Envía todo el tráfico IPv6 por la
                                interfaz del túnel
```

4.2.6.4 Configuración de protocolos de enrutamiento

```
REDFISI(config)#router bgp 10100

                                Configuración del protocolo
                                BGP.(10100)Número autónomo (ASN) que
                                identifica la configuración de BGP, en
                                este caso seleccionamos al azar
                                cualquier número. Para aplicaciones
                                sobre redes públicas debe ser un
                                número registrado

REDFISI(config-router)#no bgp default ipv4-unicast

                                Deshabilita el envío de paquetes IPv4

REDFISI(config-router)#bgp log-neighbor-changes

                                Habilita el log de los cambios de los
                                vecinos

REDFISI(config-router)#no synchronization

                                Deshabilita el diálogo entre los
                                protocolos internos y externos

REDFISI(config-router)#no auto-summary

REDFISI(config-router)#neighbor ::65.172.180.65 remote-as 20200

                                Especifica la dirección Ipv4-
                                Compatible del vecino y el número
                                autónomo (ASN) del router del otro
                                extremo

REDFISI(config-router)#address-family ipv6

                                Especifica el conjunto de comandos
                                pertenecientes a IPv6 para el
                                protocolo BGP

REDFISI(config-router-af)#neighbor ::65.172.180.65 activate

                                Habilita la dirección del vecino
```

```
REDFISI(config-router-af)#neighbor ::65.172.180.65 next-hop-self

                                Indica el siguiente salto

REDFISI(config-router-af)#bgp redistribute-internal

                                Autoriza la redistribución de rutas
                                internas de BGP (además de las rutas
                                externas BGP)en protocolos que tienen
                                habilitada la redistribución de ruteo
                                BGP.

REDFISI(config-router-af)#network FEC1:0:0:AB::/64

                                Especifica la red por la cual se
                                enviará el tráfico

REDFISI(config-router-af)#exit-address-family

REDFISI(config-router)#exit

REDFISI(config)#ipv6 router rip tesis

                                Este comando sirve para habilitar el
                                proceso de asignación de ruta del
                                protocolo RIP, va acompañado de una
                                palabra que describe dicho proceso.

REDFISI(config-router)#redistribute bgp 10100

                                Este comando se lo utiliza para
                                redistribuir las rutas IPv6 de un
                                dominio a otro.
```

4.2.6.5 Router: REDESPE

```
Router#config terminal

                                Configuración de terminal

Router(config)#hostname REDESPE

                                Nombre del Router
```

4.2.6.6 Configuración de interfases

```
REDESPE(config)#interface Ethernet0
```

```
                                Configuración de la interfase Ethernet
                                que poseerá direcciones IPv4 e IPv6

REDESPE(config-if)#ip address 10.10.1.2 255.255.255.0

                                Asigna la dirección IPv4 y la máscara
                                de subred

REDESPE(config-if)#ipv6 enable

                                Habilita IPv6

REDESPE(config-if)#ipv6 address FEC0::CD:0050:54FF:FE80:3D30/64

                                Asigna la dirección IPv6

REDESPE(config-if)#exit

REDESPE(config)#interface Serial 0

                                Configuración de la interfaz Serial

REDESPE(config-if)#ip address 65.172.180.65 255.255.255.252

                                Asigna la dirección IPv4

REDESPE(config-if)#clock rate 2000000

                                Establece la velocidad de reloj del
                                puerto, ya que este actúa como DCE.

REDESPE(config-if)#exit

REDESPE(config)#interface Tunnel 0

                                Configuración de la interfaz del túnel

REDESPE(config-if)#no ip address

                                Deshabilita IPv4

REDESPE(config-if)#ipv6 enable

                                Habilita IPv6

REDESPE(config-if)#ipv6 rip tesis enable

                                Habilita el protocolo de enrutamiento
                                RIP denominado tesis
```

```
REDESPE(config-if)#ipv6 address FEC0::1:1/126

                                Asigna la dirección IPv6 al túnel

REDESPE(config-if)#tunnel source Serial0

                                Establece el origen del túnel

REDESPE(config-if)#tunnel mode auto-tunnel

                                Establece el modo del túnel automático

REDESPE(config-if)#no shutdown

                                Habilita administrativamente el Puerto

REDESPE(config-if)#exit
```

4.2.6.7 Configuración de rutas estáticas

```
REDESPE(config)#ipv6 unicast-routing

                                Habilita el ruteo IPv6

REDESPE(config)#ip route 192.188.1.0 255.255.255.252 65.172.180.66

                                Envía todo el tráfico IPv4 hacia el
                                router INTERNET

REDESPE(config)#ipv6 route ::/0 Tunnel0

                                Envía todo el tráfico IPv6 por la
                                interfaz del túnel
```

4.2.6.8 Configuración de protocolos de enrutamiento

```
REDESPE(config)#router bgp 10100

                                Configuración del protocolo
                                BGP.(10100)Número autónomo (ASN) que
                                identifica la configuración de BGP, en
                                este caso seleccionamos al azar
                                cualquier número. Para aplicaciones
                                sobre redes públicas debe ser un
                                número registrado
```

```
REDESPE(config-router)#no bgp default ipv4-unicast

                                Deshabilita el envío de paquetes IPv4

REDESPE(config-router)#bgp log-neighbor-changes

                                Habilita el log de los cambios de los
                                vecinos

REDESPE(config-router)#no synchronization

REDESPE(config-router)#no auto-summary

REDESPE(config-router)#bgp redistribute-internal

                                Autoriza la redistribución de rutas
                                internas de BGP (además de las rutas
                                externas BGP)en protocolos que tienen
                                habilitada la redistribución de ruteo
                                BGP.

REDESPE(config-router)#neighbor ::192.188.1.1 remote-as 10100

                                Especifica la dirección Ipv4-
                                Compatible del vecino y el número
                                autónomo (ASN) del router del otro
                                extremo

REDESPE(config-router)#address-family ipv6

                                Especifica el conjunto de comandos
                                pertenecientes a IPv6 para el
                                protocolo BGP

REDESPE(config-router-af)#neighbor ::192.188.1.1 activate

                                Habilita la dirección del vecino

REDESPE(config-router-af)#neighbor ::192.188.1.1 next-hop-self

                                Indica el siguiente salto

REDESPE(config-router-af)#network FEC0:0:0:CD::/64

                                Especifica la red por la cual se
                                enviará el tráfico

REDESPE(config-router-af)#exit-address-family

REDESPE(config-router)#exit

REDESPE(config)#ipv6 router rip tesis
```

```
                                    Este comando sirve para habilitar el
                                    proceso de asignación de ruta del
                                    protocolo RIP, va acompañado de una
                                    palabra que describe dicho proceso.
REDESPE(config-router)#redistribute bgp 10100
                                    Este comando se lo utiliza para
                                    redistribuir las rutas IPv6 de un
                                    dominio a otro.
```

4.2.6.9 Router: INTERNET

```
Router#config terminal
                                    Configuración de terminal
Router(config)#hostname INTERNET
                                    Nombre del Router
```

4.2.6.10 Configuración de interfases

```
INTERNET(config)#interface Serial0
                                    Configuración de la interfaz Serial0
INTERNET(config-if)#ip address 65.172.180.66 255.255.255.0
                                    Asigna la dirección IPv4 y la máscara
                                    de subred
INTERNET(config-if)#exit
INTERNET(config)#interface Serial1
                                    Configuración de la interfaz Serial1
INTERNET(config-if)#ip address 192.188.1.2 255.255.255.252
                                    Asigna la dirección IPv4 y la máscara
                                    de subred
INTERNET(config-if)#exit
                                    Envía todo el tráfico IPv6 por la
                                    interfaz del túnel
```

4.2.7 Pruebas de Funcionamiento del Túnel Manual

- Verificación de la funcionalidad del túnel realizando ping de un router a otro.

```
redespe#ping ipv6 fec1::ab:0010:7bff:fee8:fee8:120e
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC1::AB:2D:BDFF:FEBE:BFC8, timeout
is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
140/172/216 ms
```

```
redfisi#ping ipv6 fec0::cd:0050:54ff:fe80:3d30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::CD:E0:B0FF:FE5A:9E8F, timeout
is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
132/135/136 ms
```

- Ping de un router a un host

```
redespe#ping ipv6 fec1::ab:211:25ff:fe67:5a57
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC1::AB:2E0:7DFF:FEAB:C944, timeout
is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
140/163/216 ms
```

```
redfisi#ping ipv6 fec0::cd:202:3fff:fe8b:652a
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::CD:202:3FFF:FE8B:652A, timeout
is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
136/136/136 ms
```

- Ping entre hosts

Figura 4.2: Ping Host en Red 1 → Host en Red 2

Figura 4.3: Ping Host en Red 2 → Host en Red 1

- Ping desde el host a los routers

Figura 4.4: Host en Red 2 → Router REDESPE

Figura 4.5: Host en Red 1 → Router REDFISI

- Tracert entre hosts

Figura 4.6: Desde host en Red 2 → host en Red 1

Figura 4.7: Desde host en Red 1 → host en Red 2

- Tracert con la opción -R que muestra la ruta de ida y regreso de un paquete desde una máquina de la Red 1 a otra en la Red 2

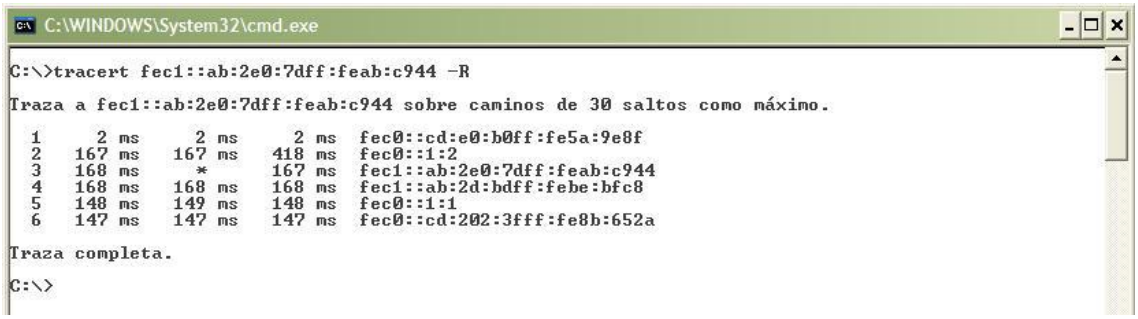


Figura 4.8: Desde host en Red 2 → host en Red 1

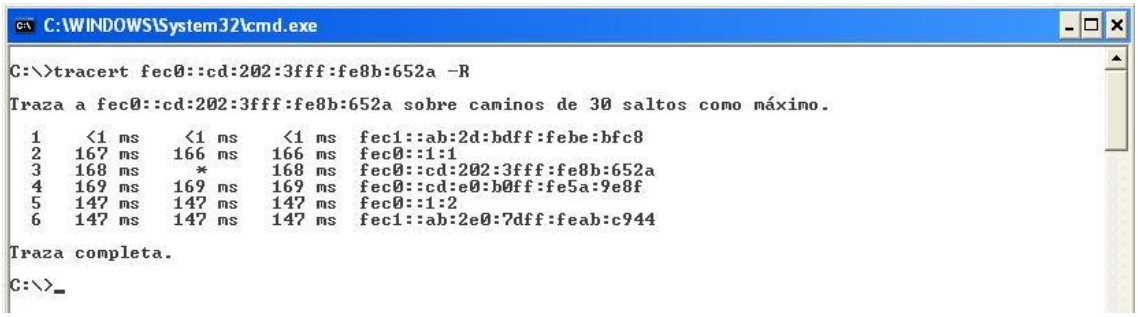


Figura 4.9: Desde host en Red 1 → host en Red 2

Por medio del comando show en los routers se pueden verificar:

- Túnel en REDFISI

```
redfisi#show interfaces tunnel0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  Description: Tunel Manual
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 192.188.1.1, destination 65.172.180.65
  Tunnel protocol/transport IPv6/IP, key disabled, sequencing disabled
  Tunnel TTL 255
  Checksumming of packets disabled,  fast tunneling enabled
  Last input 00:08:12, output 00:00:50, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     310 packets input, 37456 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     317 packets output, 31916 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
```

```
     0 output buffer failures, 0 output buffers swapped out
```

- Túnel en REDESPE

```
redespe#show interfaces tunnel0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 65.172.180.65, destination 192.188.1.1
  Tunnel protocol/transport IPv6/IP, key disabled, sequencing disabled
  Tunnel TTL 255
  Checksumming of packets disabled,  fast tunneling enabled
  Last input never, output 00:01:16, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     6 packets output, 520 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

- Interfases de REDFISI

```
redfisi#show ipv6 interface
FastEthernet0 is up, line protocol is up
  IPv6      is      enabled,      link-local      address      is
FE80::20D:BDFF:FEBE:BFC8
  Global unicast address(es):
    FEC1::AB:2D:BDFF:FEBE:BFC8, subnet is FEC1:0:0:AB::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FFBE:BFC8
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
Tunnel0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::C0BC:101
  Description: Tunel Manual
  Global unicast address(es):
    FEC0::1:2, subnet is FEC0::1:0/126
  Joined group address(es):
```

```
    FF02::1
    FF02::2
    FF02::1:FF01:2
    FF02::1:FFBC:101
  MTU is 1480 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  Hosts use stateless autoconfig for addresses.
```

- Interfases de REDESPE

```
redespe#show ipv6 interface
Ethernet0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::2E0:B0FF:FE5A:9E8F
  Description: red local FEC0::3
  Global unicast address(es):
    FEC0::CD:E0:B0FF:FE5A:9E8F, subnet is FEC0:0:0:CD::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF5A:9E8F
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
Tunnel0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::41AC:B441
  Global unicast address(es):
    FEC0::1:1, subnet is FEC0::1:0/126
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF01:1
    FF02::1:FFAC:B441
  MTU is 1480 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  Hosts use stateless autoconfig for addresses.
redespe#
```

- Rutas de REDFISI

```
redfisi#show ipv6 route
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS
```

```
summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF
ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S   ::/0 [1/0]
     via ::, Tunnel0
L   FE80::/10 [0/0]
     via ::, Null0
C   FEC0::1:0/126 [0/0]
     via ::, Tunnel0
L   FEC0::1:2/128 [0/0]
     via ::, Tunnel0
C   FEC1:0:0:AB::/64 [0/0]
     via ::, FastEthernet0
L   FEC1::AB:2D:BDFF:FEBE:BFC8/128 [0/0]
     via ::, FastEthernet0
L   FF00::/8 [0/0]
     via ::, Null0
redfisi#
```

- Rutas de REDESPE

```
redespe#show ipv6 route
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
L   FE80::/10 [0/0]
     via ::, Null0
L   FEC0::1:1/128 [0/0]
     via ::, Tunnel0
C   FEC0::1:0/126 [0/0]
     via ::, Tunnel0
L   FEC0::CD:E0:B0FF:FE5A:9E8F/128 [0/0]
     via ::, Ethernet0
C   FEC0:0:0:CD::/64 [0/0]
     via ::, Ethernet0
L   FF00::/8 [0/0]
     via ::, Null0
S   ::/0 [1/0]
     via FEC0::1:2, Null
```

- Protocolos de Ruteo de REDFISI

```
redfisi#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
```

- Protocolos de Ruteo de REDESPE

```
redespe#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
```

- Descubrimiento de Vecinos de REDFISI

```
redfisi#show ipv6 neighbors
IPv6 Address                          Age Link-layer Addr State Interface
FE80::2E0:7DFF:FEAB:C944              4 00e0.7dab.c944  STALE Fa0
FEC1::AB:2E0:7DFF:FEAB:C944           4 00e0.7dab.c944  STALE Fa0
```

- Descubrimiento de Vecinos de REDESPE

```
redespe#show ipv6 neighbors
IPv6 Address                          Age Link-layer Addr State Interface
FEC0::CD:202:3FFF:FE8B:652A           16 0002.3f8b.652a  STALE Et0
FE80::202:3FFF:FE8B:652A              16 0002.3f8b.652a  STALE Et0
```

Por medio del comando debug se activa la captura de los paquetes que circulan por la red, al momento de hacer un ping entre redes IPv6.

- Paquetes IPv6 en REDFISI

```
redfisi#debug ipv6 packet detail
IPv6 unicast packet debugging is on (detailed)
redfisi#
*Mar     1   00:37:59.443:   IPV6:   source   FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 00:37:59.443:        dest FEC0::CD:202:3FFF:FE8B:652A (Tunnel0)
*Mar  1 00:37:59.447:        traffic class 0, flow 0x0, len 80+14, prot
58, hops 127, forwarding
*Mar     1   00:37:59.547:   IPV6:   source   FEC0::CD:202:3FFF:FE8B:652A
(Tunnel0)
*Mar    1  00:37:59.547:              dest  FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 00:37:59.547:        traffic class 0, flow 0x0, len 80+20, prot
58, hops 62, forwarding
*Mar     1   00:38:00.451:   IPV6:   source   FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 00:38:00.451:        dest FEC0::CD:202:3FFF:FE8B:652A (Tunnel0)
*Mar  1 00:38:00.455:        traffic class 0, flow 0x0, len 80+14, prot
58, hops 127, forwarding
*Mar     1   00:38:00.771:   IPV6:   source   FEC0::CD:202:3FFF:FE8B:652A
(Tunnel0)
*Mar    1  00:38:00.775:              dest  FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 00:38:00.775:        traffic class 0, flow 0x0, len 80+20, prot
58, hops 62, forwarding
*Mar     1   00:38:01.455:   IPV6:   source   FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 00:38:01.455:        dest FEC0::CD:202:3FFF:FE8B:652A (Tunnel0)
*Mar  1 00:38:01.455:        traffic class 0, flow 0x0, len 80+14, prot
58, hops 127, forwarding
*Mar     1   00:38:01.555:   IPV6:   source   FEC0::CD:202:3FFF:FE8B:652A
(Tunnel0)
*Mar    1  00:38:01.555:              dest  FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 00:38:01.555:        traffic class 0, flow 0x0, len 80+20, prot
58, hops 62, forwarding
```

```
*Mar     1   00:38:02.455:   IPV6:   source   FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 00:38:02.455:       dest FEC0::CD:202:3FFF:FE8B:652A (Tunnel0)
*Mar  1 00:38:02.455:       traffic class 0, flow 0x0, len 80+14, prot
58, hops 127, forwarding
*Mar     1   00:38:02.555:   IPV6:   source   FEC0::CD:202:3FFF:FE8B:652A
(Tunnel0)
*Mar    1  00:38:02.559:             dest  FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 00:38:02.559:       traffic class 0, flow 0x0, len 80+20, prot
58, hops 62, forwarding
*Mar      1   00:38:04.219:   IPV6:   source   FE80::2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 00:38:04.219:       dest FE80::20D:BDFF:FEBE:BFC8
*Mar  1 00:38:04.219:       traffic class 0, flow 0x0, len 72+14, prot
58, hops 255, forward to ulp
*Mar  1 00:38:04.219: IPV6: source FE80::20D:BDFF:FEBE:BFC8 (local)
*Mar     1  00:38:04.219:               dest  FE80::2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar   1 00:38:04.219:         traffic class 224, flow 0x0, len 64+16,
prot 58, hops 255, originating
*Mar  1 00:38:04.219: IPv6: Sending on FastEthernet0
*Mar  1 00:38:04.547: IPV6: source FE80::20D:BDFF:FEBE:BFC8 (local)
*Mar    1  00:38:04.547:             dest  FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar   1 00:38:04.547:         traffic class 224, flow 0x0, len 72+8,
prot 58, hops 255, originating
*Mar  1 00:38:04.547: IPv6: Sending on FastEthernet0
*Mar     1   00:38:04.547:   IPV6:   source   FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 00:38:04.547:       dest FE80::20D:BDFF:FEBE:BFC8
*Mar  1 00:38:04.547:       traffic class 0, flow 0x0, len 72+14, prot
58, hops 255, forward to ulp
*Mar  1 00:38:09.219: IPV6: source FE80::20D:BDFF:FEBE:BFC8 (local)
*Mar     1  00:38:09.219:               dest  FE80::2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar   1 00:38:09.219:         traffic class 224, flow 0x0, len 72+8,
prot 58, hops 255, originating
*Mar  1 00:38:09.219: IPv6: Sending on FastEthernet0
*Mar      1   00:38:09.219:   IPV6:   source   FE80::2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 00:38:09.219:       dest FE80::20D:BDFF:FEBE:BFC8
*Mar  1 00:38:09.219:       traffic class 0, flow 0x0, len 72+14, prot
58, hops 255, forward to ulp
```

- Paquetes IPv6 en REDESPE

```
redespe#debug ipv6 packet detail
IPv6 unicast packet debugging is on (detailed)
redespe#
01:43:48: IPV6: source FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
01:43:48:       dest FEC1::AB:2E0:7DFF:FEAB:C944 (Tunnel0)
01:43:48:         traffic class 0, flow 0x0, len 80+14, prot 58, hops
63, forwarding
01:43:48: IPV6: source FEC1::AB:2E0:7DFF:FEAB:C944 (Tunnel0)
01:43:48:       dest FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
01:43:48:         traffic class 0, flow 0x0, len 80+20, prot 58, hops
62, forwarding
01:43:49: IPV6: source FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
```

```
01:43:49:       dest FEC1::AB:2E0:7DFF:FEAB:C944 (Tunnel0)
01:43:49:        traffic class 0, flow 0x0, len 80+14, prot 58, hops
63, forwarding
01:43:49: IPV6: source FEC1::AB:2E0:7DFF:FEAB:C944 (Tunnel0)
01:43:49:       dest FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
01:43:49:        traffic class 0, flow 0x0, len 80+20, prot 58, hops
62, forwarding
01:43:50: IPV6: source FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
01:43:50:       dest FEC1::AB:2E0:7DFF:FEAB:C944 (Tunnel0)
01:43:50:        traffic class 0, flow 0x0, len 80+14, prot 58, hops
63, forwarding
01:43:50: IPV6: source FEC1::AB:2E0:7DFF:FEAB:C944 (Tunnel0)
01:43:50:       dest FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
01:43:50:        traffic class 0, flow 0x0, len 80+20, prot 58, hops
62, forwarding
01:43:51: IPV6: source FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
01:43:51:       dest FEC1::AB:2E0:7DFF:FEAB:C944 (Tunnel0)
01:43:51:        traffic class 0, flow 0x0, len 80+14, prot 58, hops
63, forwarding
01:43:51: IPV6: source FEC1::AB:2E0:7DFF:FEAB:C944 (Tunnel0)
01:43:51:       dest FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
01:43:51:        traffic class 0, flow 0x0, len 80+20, prot 58, hops
62, forwarding
01:43:52: IPV6: source FE80::202:3FFF:FE8B:652A (Ethernet0)
01:43:52:       dest FE80::2E0:B0FF:FE5A:9E8F
01:43:52:        traffic class 0, flow 0x0, len 72+14, prot 58, hops
255, forward to ulp
01:43:52: IPV6: source FE80::2E0:B0FF:FE5A:9E8F (local)
01:43:52:       dest FE80::202:3FFF:FE8B:652A (Ethernet0)
01:43:52:        traffic class 224, flow 0x0, len 64+16, prot 58, hops
255, originating
01:43:52: IPv6: Sending on Ethernet0
01:43:53: IPV6: source FE80::2E0:B0FF:FE5A:9E8F (local)
01:43:53:       dest FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
01:43:53:        traffic class 224, flow 0x0, len 72+8, prot 58, hops
255, originating
01:43:53: IPv6: Sending on Ethernet0
01:43:53: IPV6: source FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
01:43:53:       dest FE80::2E0:B0FF:FE5A:9E8F
01:43:53:        traffic class 0, flow 0x0, len 72+14, prot 58, hops
255, forward to ulp
01:43:57: IPV6: source FE80::2E0:B0FF:FE5A:9E8F (local)
01:43:57:       dest FE80::202:3FFF:FE8B:652A (Ethernet0)
01:43:57:        traffic class 224, flow 0x0, len 72+8, prot 58, hops
255, originating
01:43:57: IPv6: Sending on Ethernet0
01:43:57: IPV6: source FE80::202:3FFF:FE8B:652A (Ethernet0)
01:43:57:       dest FE80::2E0:B0FF:FE5A:9E8F
01:43:57:        traffic class 0, flow 0x0, len 72+14, prot 58, hops
255, forward to ulp
```

- Paquetes ICMP en REDFISI

```
redfisi#debug ipv6 icmp
ICMP packet debugging is on
redfisi#
*Mar     1   00:40:11.903:   ICMPv6:   Received   ICMPv6   packet   from
FE80::2E0:7DFF:FEAB:C944, type 135
*Mar     1   00:40:12.483:   ICMPv6:   Received   ICMPv6   packet   from
FEC1::AB:2E0:7DFF:FEAB:C944, type 136
```

```
*Mar     1   00:40:16.903:   ICMPv6:   Received   ICMPv6   packet   from
FE80::2E0:7DFF:FEAB:C944, type 136
```

- Paquetes ICMP en REDESPE

```
redespe#debug ipv6 icmp
ICMP packet debugging is on
redespe#
01:47:20:      ICMPv6:      Received      ICMPv6      packet      from
FE80::202:3FFF:FE8B:652A, type 135
01:47:21:      ICMPv6:      Received      ICMPv6      packet      from
FEC0::CD:202:3FFF:FE8B:652A, type 136
01:47:25:      ICMPv6:      Received      ICMPv6      packet      from
FE80::202:3FFF:FE8B:652A, type 136
```

- Descubrimiento de Vecinos en REDFISI

```
redfisi#debug ipv6 nd
ICMP Neighbor Discovery events debugging is on
redfisi#
*Mar         1     00:41:22.171:     ICMPv6-ND:     STALE     ->     DELAY:
FEC1::AB:2E0:7DFF:FEAB:C944
*Mar         1     00:41:27.011:     ICMPv6-ND:     Received     NS     for
FE80::20D:BDFF:FEBE:BFC8          on          FastEthernet0          from
FE80::2E0:7DFF:FEAB:C944
*Mar         1     00:41:27.015:     ICMPv6-ND:     Sending     NA     for
FE80::20D:BDFF:FEBE:BFC8 on FastEthernet0
*Mar         1     00:41:27.015:     ICMPv6-ND:     STALE     ->     DELAY:
FE80::2E0:7DFF:FEAB:C944
*Mar         1     00:41:27.171:     ICMPv6-ND:     DELAY     ->     PROBE:
FEC1::AB:2E0:7DFF:FEAB:C944
*Mar         1     00:41:27.171:     ICMPv6-ND:     Sending     NS     for
FEC1::AB:2E0:7DFF:FEAB:C944 on FastEthernet0
*Mar         1     00:41:27.171:     ICMPv6-ND:     Received     NA     for
FEC1::AB:2E0:7DFF:FEAB:C944         on         FastEthernet0         from
FEC1::AB:2E0:7DFF:FEAB:C944
*Mar         1     00:41:27.171:     ICMPv6-ND:     PROBE     ->     REACH:
FEC1::AB:2E0:7DFF:FEAB:C944
*Mar         1     00:41:32.015:     ICMPv6-ND:     DELAY     ->     PROBE:
FE80::2E0:7DFF:FEAB:C944
*Mar         1     00:41:32.015:     ICMPv6-ND:     Sending     NS     for
FE80::2E0:7DFF:FEAB:C944 on FastEthernet0
*Mar         1     00:41:32.015:     ICMPv6-ND:     Received     NA     for
FE80::2E0:7DFF:FEAB:C944          on          FastEthernet0          from
FE80::2E0:7DFF:FEAB:C944
*Mar         1     00:41:32.015:     ICMPv6-ND:     PROBE     ->     REACH:
FE80::2E0:7DFF:FEAB:C944
```

- Descubrimiento de Vecinos en REDESPE

```
redespe#debug ipv6 nd
ICMP Neighbor Discovery events debugging is on
redespe#
01:49:22: ICMPv6-ND: STALE -> DELAY: FEC0::CD:202:3FFF:FE8B:652A
01:49:27:  ICMPv6-ND:  Received  NS  for  FE80::2E0:B0FF:FE5A:9E8F  on
Ethernet0 from FE80::202:3FFF:FE8B:652A
01:49:27:  ICMPv6-ND:  Sending  NA  for  FE80::2E0:B0FF:FE5A:9E8F  on
Ethernet0
```

```
01:49:27: ICMPv6-ND: STALE -> DELAY: FE80::202:3FFF:FE8B:652A
01:49:27: ICMPv6-ND: DELAY -> PROBE: FEC0::CD:202:3FFF:FE8B:652A
01:49:27: ICMPv6-ND: Sending NS for FEC0::CD:202:3FFF:FE8B:652A on
Ethernet0
01:49:27: ICMPv6-ND: Received NA for FEC0::CD:202:3FFF:FE8B:652A on
Ethernet0 from FEC0::CD:202:3FFF:FE8B:652A
01:49:27: ICMPv6-ND: PROBE -> REACH: FEC0::CD:202:3FFF:FE8B:652A
01:49:32: ICMPv6-ND: DELAY -> PROBE: FE80::202:3FFF:FE8B:652A
01:49:32: ICMPv6-ND: Sending NS for FE80::202:3FFF:FE8B:652A on
Ethernet0
01:49:32: ICMPv6-ND: Received NA for FE80::202:3FFF:FE8B:652A on
Ethernet0 from FE80::202:3FFF:FE8B:652A
01:49:32: ICMPv6-ND: PROBE -> REACH: FE80::202:3FFF:FE8B:652A
```

4.2.8 Pruebas de Funcionamiento del Túnel Automático

- Se verifica la funcionalidad del túnel realizando un ping de un router a otro

```
redespe#ping ipv6 fec1::ab:0010:7bff:fee8:fee8:120e
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC1::AB:2D:BDFF:FEBE:BFC8, timeout
is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
140/172/216 ms
```

```
redfisi#ping ipv6 fec0::cd:0050:54ff:fe80:3d30

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::CD:E0:B0FF:FE5A:9E8F, timeout
is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
132/135/136 ms
```

- Se realiza un ping de un router a un host

```
redespe#ping ipv6 fec1::ab:211:25ff:fe67:5a57
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC1::AB:2E0:7DFF:FEAB:C944, timeout
is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
140/163/216 ms
```

```
redfisi#ping ipv6 fec0::cd:202:3fff:fe8b:652a

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::CD:202:3FFF:FE8B:652A, timeout
is 2 seconds:
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max =
136/136/136 ms
```

- Se realiza un ping entre hosts

Figura 4.10: Host en Red 2 → Host en Red 1

Figura 4.11: Host en Red 1 → Host en Red 2

- Se realiza un desde el host a los routers

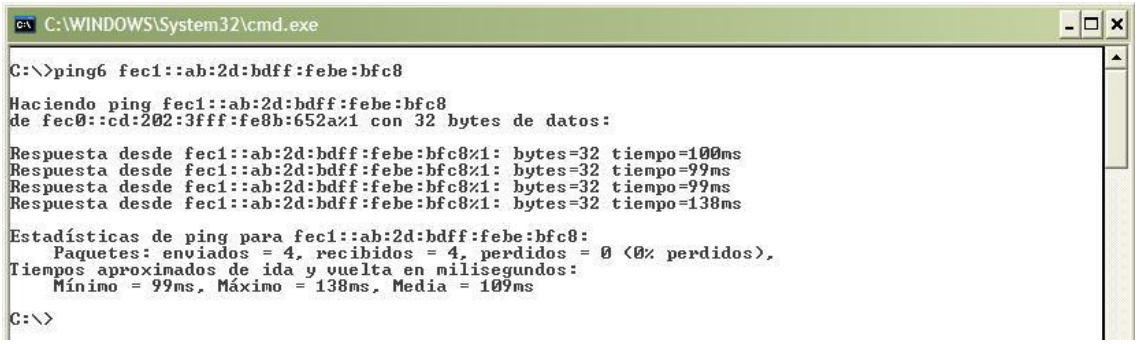


Figura 4.12: Host en Red 2 → Router REDFISI

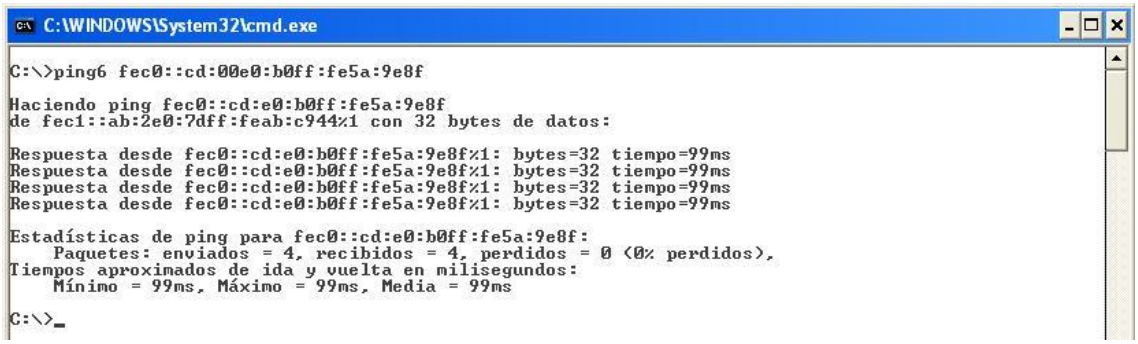


Figura 4.13: Host en Red 1 → Router REDESPE

- Se realiza un tracert entre hosts

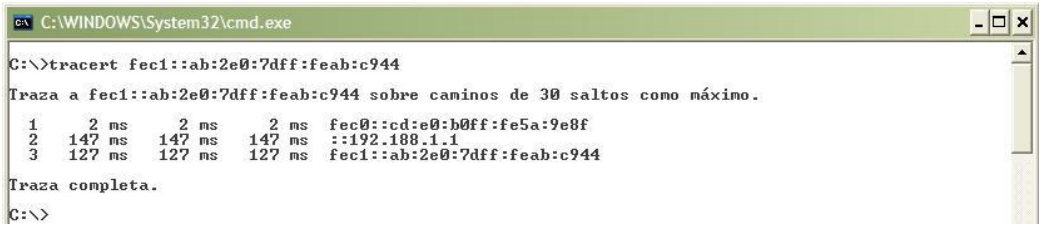


Figura 4.14: Desde host en Red 2 → host en Red 1

Figura 4.15: Desde host en Red 1 → host en Red 2

- Tracert con la opción -R que muestra la ruta de ida y regreso de un paquete desde una máquina de la Red 1 a otra en la Red 2

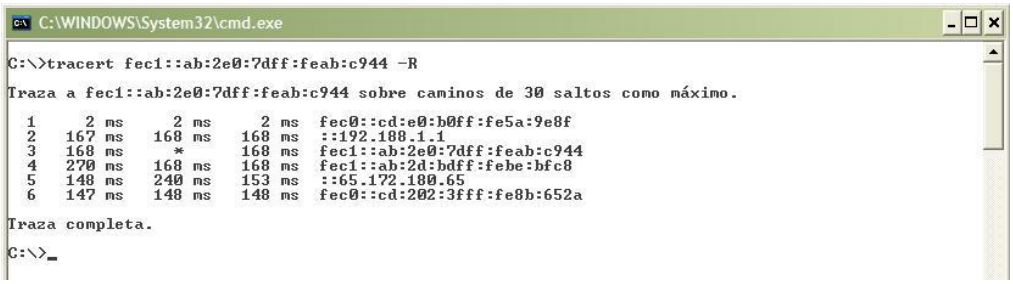


Figura 4.16: Desde host en Red 2 → host en Red 1

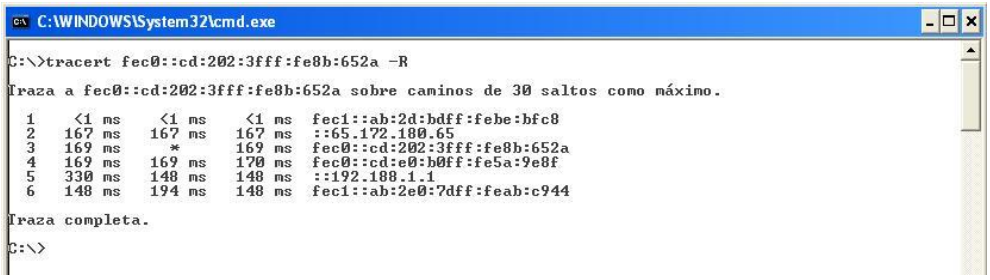


Figura 4.17: Desde host en Red 1 → host en Red 2

Por medio del comando show en los routers se pueden verificar:

- Túnel en REDFISI
- Túnel en REDESPE
- Interfases de REDFISI

```
redfisi#show ipv6 interface
FastEthernet0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20D:BDFF:FEBE:BFC8
  Global unicast address(es):
    FEC1::AB:2D:BDFF:FEBE:BFC8, subnet is FEC1:0:0:AB::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FFBE:BFC8
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
Tunnel0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::C0BC:101
```

```
  Global unicast address(es):
    ::192.188.1.1, subnet is ::/96
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::9
    FF02::1:FFBC:101
  MTU is 1480 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is not supported
  ND reachable time is 30000 milliseconds
  Hosts use stateless autoconfig for addresses.
```

- Interfases de REDESPE

```
redespe#show ipv6 interfaces
Ethernet0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::2E0:B0FF:FE5A:9E8F
  Description: red local FEC0::3
  Global unicast address(es):
    FEC0::CD:E0:B0FF:FE5A:9E8F, subnet is FEC0:0:0:CD::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF5A:9E8F
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
Tunnel0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::41AC:B441
  Global unicast address(es):
    ::65.172.180.65, subnet is ::/96
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::9
    FF02::1:FFAC:B441
  MTU is 1480 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is not supported
  ND reachable time is 30000 milliseconds
  Hosts use stateless autoconfig for addresses.
```

- Rutas de REDFISI

```
redfisi#show ipv6 route
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
```

```
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS
summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF
ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
C   ::/96 [0/0]
     via ::, Tunnel0
L   ::192.188.1.1/128 [0/0]
     via ::, Tunnel0
L   FE80::/10 [0/0]
     via ::, Null0
B   FEC0:0:0:CD::/64 [20/0]
     via ::65.172.180.65
C   FEC1:0:0:AB::/64 [0/0]
     via ::, FastEthernet0
L   FEC1::AB:2D:BDFF:FEBE:BFC8/128 [0/0]
     via ::, FastEthernet0
L   FF00::/8 [0/0]
     via ::, Null0
```

- Rutas de REDESPE

```
redespe#show ipv6 route
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
L   ::65.172.180.65/128 [0/0]
     via ::, Tunnel0
C   ::/96 [0/0]
     via ::, Tunnel0
L   FE80::/10 [0/0]
     via ::, Null0
L   FEC0::CD:E0:B0FF:FE5A:9E8F/128 [0/0]
     via ::, Ethernet0
C   FEC0:0:0:CD::/64 [0/0]
     via ::, Ethernet0
B   FEC1:0:0:AB::/64 [20/0]
     via ::192.188.1.1, Tunnel0
L   FF00::/8 [0/0]
     via ::, Null0
```

- Protocolos de Ruteo de REDFISI

```
redfisi#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "rip tesis"
  Interfaces:
    Tunnel0
  Redistribution:
    Redistributing protocol bgp 10100
IPv6 Routing Protocol is "bgp 10100"
  IGP synchronization is disabled
  Redistribution:
    None
```

```
  Neighbor(s):
    Address FiltIn FiltOut Weight RoutemapIn RoutemapOut
    ::65.172.180.65
```

- Protocolos de Ruteo de REDESPE

```
redespe#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "rip tesis"
  Interfaces:
    Tunnel0
  Redistribution:
    Redistributing protocol bgp 20200
IPv6 Routing Protocol is "bgp 20200"
  IGP synchronization is disabled
  Redistribution:
    None
  Neighbor(s):
    Address FiltIn FiltOut Weight RoutemapIn RoutemapOut
    ::192.188.1.1
```

- Descubrimiento de Vecinos de REDFISI

```
redfisi#show ipv6 neighbors
IPv6 Address                          Age Link-layer Addr State Interface
FE80::2E0:7DFF:FEAB:C944               5 00e0.7dab.c944  STALE Fa0
FEC1::AB:2E0:7DFF:FEAB:C944            5 00e0.7dab.c944  STALE Fa0
FE80::202:3FFF:FE8B:652A              59 0002.3f8b.652a  STALE Fa0
::65.172.180.65                        0 65.172.180.65   STALE Tu0
redfisi#
```

- Descubrimiento de Vecinos de REDESPE

```
redespe#show ipv6 neighbors
IPv6 Address                           Age Link-layer Addr State Interface
::192.188.1.1                           4 192.188.1.1     STALE Tu0
FE80::202:3FFF:FE8B:652A               10 0002.3f8b.652a  STALE Et0
```

Por medio del comando debug se activa la captura de los paquetes que circulan por la red, al momento de hacer un ping entre redes IPv6.

- Paquetes IPv6 en REDFISI

```
redfisi#debug ipv6 packet detail
IPv6 unicast packet debugging is on (detailed)
redfisi#
```

```
*Mar     1   01:23:05.251:   IPV6:   source   FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 01:23:05.251:       dest FEC0::CD:202:3FFF:FE8B:652A (Tunnel0)
*Mar  1 01:23:05.251:       traffic class 0, flow 0x0, len 80+14, prot
58, hops 127, forwarding
*Mar     1   01:23:05.355:   IPV6:   source   FEC0::CD:202:3FFF:FE8B:652A
(Tunnel0)
*Mar    1  01:23:05.355:             dest  FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 01:23:05.355:       traffic class 0, flow 0x0, len 80+20, prot
58, hops 62, forwarding
*Mar     1   01:23:06.271:   IPV6:   source   FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 01:23:06.271:       dest FEC0::CD:202:3FFF:FE8B:652A (Tunnel0)
*Mar  1 01:23:06.271:       traffic class 0, flow 0x0, len 80+14, prot
58, hops 127, forwarding
*Mar     1   01:23:06.371:   IPV6:   source   FEC0::CD:202:3FFF:FE8B:652A
(Tunnel0)
*Mar    1  01:23:06.371:             dest  FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 01:23:06.375:       traffic class 0, flow 0x0, len 80+20, prot
58, hops 62, forwarding
*Mar     1   01:23:07.271:   IPV6:   source   FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 01:23:07.271:       dest FEC0::CD:202:3FFF:FE8B:652A (Tunnel0)
*Mar  1 01:23:07.271:       traffic class 0, flow 0x0, len 80+14, prot
58, hops 127, forwarding
*Mar     1   01:23:07.375:   IPV6:   source   FEC0::CD:202:3FFF:FE8B:652A
(Tunnel0)
*Mar    1  01:23:07.375:             dest  FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 01:23:07.375:       traffic class 0, flow 0x0, len 80+20, prot
58, hops 62, forwarding
*Mar     1   01:23:08.275:   IPV6:   source   FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 01:23:08.275:       dest FEC0::CD:202:3FFF:FE8B:652A (Tunnel0)
*Mar  1 01:23:08.275:       traffic class 0, flow 0x0, len 80+14, prot
58, hops 127, forwarding
*Mar     1   01:23:08.375:   IPV6:   source   FEC0::CD:202:3FFF:FE8B:652A
(Tunnel0)
*Mar    1  01:23:08.375:             dest  FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 01:23:08.375:       traffic class 0, flow 0x0, len 80+20, prot
58, hops 62, forwarding
*Mar      1   01:23:09.975:   IPV6:   source   FE80::2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 01:23:09.975:       dest FE80::20D:BDFF:FEBE:BFC8
*Mar  1 01:23:09.975:       traffic class 0, flow 0x0, len 72+14, prot
58, hops 255, forward to ulp
*Mar  1 01:23:09.979: IPV6: source FE80::20D:BDFF:FEBE:BFC8 (local)
*Mar     1   01:23:09.979:               dest  FE80::2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 01:23:09.979:    traffic class 224, flow 0x0, len 64+16, prot
58, hops 255, originating
*Mar  1 01:23:09.979: IPv6: Sending on FastEthernet0
*Mar  1 01:23:10.355: IPV6: source FE80::20D:BDFF:FEBE:BFC8 (local)
*Mar    1  01:23:10.355:             dest  FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 01:23:10.355:     traffic class 224, flow 0x0, len 72+8, prot
58, hops 255, originating
*Mar  1 01:23:10.355: IPv6: Sending on FastEthernet0
```

```
*Mar     1   01:23:10.355:   IPV6:   source   FEC1::AB:2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 01:23:10.355:       dest FE80::20D:BDFF:FEBE:BFC8
*Mar  1 01:23:10.355:      traffic class 0, flow 0x0, len 72+14, prot
58, hops 255, forward to ulp
*Mar  1 01:23:12.839: IPV6: source FE80::C0BC:101 (local)
*Mar  1 01:23:12.839:       dest FF02::9 (Tunnel0)
*Mar  1 01:23:12.839:        traffic class 224, flow 0x0, len 92+1388,
prot 17, hops 255, originating
*Mar  1 01:23:12.839: IPv6: Encapsulation failed
*Mar  1 01:23:14.979: IPV6: source FE80::20D:BDFF:FEBE:BFC8 (local)
*Mar     1   01:23:14.979:               dest   FE80::2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 01:23:14.979:      traffic class 224, flow 0x0, len 72+8, prot
58, hops 255, originating
*Mar  1 01:23:14.979: IPv6: Sending on FastEthernet0
*Mar      1    01:23:14.979:    IPV6:    source    FE80::2E0:7DFF:FEAB:C944
(FastEthernet0)
*Mar  1 01:23:14.979:       dest FE80::20D:BDFF:FEBE:BFC8
*Mar  1 01:23:14.979:       traffic class 0, flow 0x0, len 72+14, prot
58, hops 255, forward to ulp
```

- Paquetes IPv6 en REDESPE

```
redespe#debug ipv6 packet detail
IPv6 unicast packet debugging is on (detailed)
redespe#
01:07:05: IPV6: source FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
01:07:05:       dest FEC1::AB:2E0:7DFF:FEAB:C944 (Tunnel0)
01:07:05:    traffic class 0, flow 0x0, len 80+14, prot 58, hops 63,
forwarding
01:07:05: IPV6: source FEC1::AB:2E0:7DFF:FEAB:C944 (Tunnel0)
01:07:05:       dest FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
01:07:05:   traffic class 0, flow 0x0, len 80+20, prot 58, hops 62,
forwarding
01:07:06: IPV6: source FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
01:07:06:       dest FEC1::AB:2E0:7DFF:FEAB:C944 (Tunnel0)
01:07:06: traffic class 0, flow 0x0, len 80+14, prot 58, hops 63,
forwarding
01:07:06: IPV6: source FEC1::AB:2E0:7DFF:FEAB:C944 (Tunnel0)
01:07:06:       dest FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
01:07:06: traffic class 0, flow 0x0, len 80+20, prot 58, hops 62,
forwarding
01:07:07: IPV6: source FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
01:07:07:       dest FEC1::AB:2E0:7DFF:FEAB:C944 (Tunnel0)
01:07:07: traffic class 0, flow 0x0, len 80+14, prot 58, hops 63,
forwarding
01:07:07: IPV6: source FEC1::AB:2E0:7DFF:FEAB:C944 (Tunnel0)
01:07:07:       dest FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
01:07:07: traffic class 0, flow 0x0, len 80+20, prot 58, hops 62,
forwarding
01:07:08: IPV6: source FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
01:07:08:       dest FEC1::AB:2E0:7DFF:FEAB:C944 (Tunnel0)
01:07:08: traffic class 0, flow 0x0, len 80+14, prot 58, hops 63,
forwarding
01:07:08: IPV6: source FEC1::AB:2E0:7DFF:FEAB:C944 (Tunnel0)
01:07:08:       dest FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
01:07:08: traffic class 0, flow 0x0, len 80+20, prot 58, hops 62,
forwarding
```

```
01:07:09: IPV6: source FE80::202:3FFF:FE8B:652A (Ethernet0)
01:07:09:       dest FE80::2E0:B0FF:FE5A:9E8F
01:07:09:    traffic class 0, flow 0x0, len 72+14, prot 58, hops 255,
forward to ulp
01:07:09: IPV6: source FE80::2E0:B0FF:FE5A:9E8F (local)
01:07:09:       dest FE80::202:3FFF:FE8B:652A (Ethernet0)
01:07:09:   traffic class 224, flow 0x0, len 64+16, prot 58, hops 255,
originating
01:07:09: IPv6: Sending on Ethernet0
01:07:10: IPV6: source FE80::2E0:B0FF:FE5A:9E8F (local)
01:07:10:       dest FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
01:07:10:    traffic class 224, flow 0x0, len 72+8, prot 58, hops 255,
originating
01:07:10: IPv6: Sending on Ethernet0
01:07:10: IPV6: source FEC0::CD:202:3FFF:FE8B:652A (Ethernet0)
01:07:10:       dest FE80::2E0:B0FF:FE5A:9E8F
01:07:10:   traffic class 0, flow 0x0, len 72+14, prot 58, hops 255,
forward to ulp
01:07:10: IPV6: source FE80::41AC:B441 (local)
01:07:10:       dest FF02::9 (Tunnel0)
01:07:10: traffic class 224, flow 0x0, len 92+1388, prot 17, hops 255,
originating
01:07:10: IPv6: Encapsulation failed
01:07:14: IPV6: source FE80::2E0:B0FF:FE5A:9E8F (local)
01:07:14:       dest FE80::202:3FFF:FE8B:652A (Ethernet0)
01:07:14: traffic class 224, flow 0x0, len 72+8, prot 58, hops 255,
originating
01:07:14: IPv6: Sending on Ethernet0
01:07:14: IPV6: source FE80::202:3FFF:FE8B:652A (Ethernet0)
01:07:14:       dest FE80::2E0:B0FF:FE5A:9E8F
01:07:14: traffic class 0, flow 0x0, len 72+14, prot 58, hops 255,
forward to ulp
```

- Paquetes ICMP en REDFISI

```
redfisi#debug ipv6 icmp
ICMP packet debugging is on
redfisi#
*Mar     1   01:24:35.099:   ICMPv6:   Received   ICMPv6   packet   from
FE80::2E0:7DFF:FEAB:C9
44, type 135
*Mar     1   01:24:35.607:   ICMPv6:   Received   ICMPv6   packet   from
FEC1::AB:2E0:7DFF:FEAB
:C944, type 136
*Mar     1   01:24:40.099:   ICMPv6:   Received   ICMPv6   packet   from
FE80::2E0:7DFF:FEAB:C9
44, type 136
redfisi#
```

- Paquetes ICMP en REDESPE

```
redespe#debug ipv6 icmp
ICMP packet debugging is on
redespe#
01:08:58:      ICMPv6:      Received      ICMPv6      packet      from
FE80::202:3FFF:FE8B:652A, type 135
01:08:59:      ICMPv6:      Received      ICMPv6      packet      from
```

```
FEC0::CD:202:3FFF:FE8B:652A, type 136
01:09:03:      ICMPv6:      Received      ICMPv6      packet      from
FE80::202:3FFF:FE8B:652A, type 136
redespe#
```

- Descubrimiento de Vecinos en REDFISI

```
redfisi#debug ipv6 nd
ICMP Neighbor Discovery events debugging is on
redfisi#
*Mar         1     01:25:40.655:     ICMPv6-ND:     STALE     ->     DELAY:
FEC1::AB:2E0:7DFF:FEAB:C944
*Mar         1     01:25:45.203:     ICMPv6-ND:     Received     NS     for
FE80::20D:BDFF:FEBE:BFC8          on          FastEthernet0          from
FE80::2E0:7DFF:FEAB:C944
*Mar         1     01:25:45.203:     ICMPv6-ND:     Sending     NA     for
FE80::20D:BDFF:FEBE:BFC8 on FastEthernet0
*Mar         1     01:25:45.203:     ICMPv6-ND:     STALE     ->     DELAY:
FE80::2E0:7DFF:FEAB:C944
*Mar         1     01:25:45.655:     ICMPv6-ND:     DELAY     ->     PROBE:
FEC1::AB:2E0:7DFF:FEAB:C944
*Mar         1     01:25:45.655:     ICMPv6-ND:     Sending     NS     for
FEC1::AB:2E0:7DFF:FEAB:C944 on FastEthernet0
*Mar         1     01:25:45.655:     ICMPv6-ND:     Received     NA     for
FEC1::AB:2E0:7DFF:FEAB:C944         on         FastEthernet0         from
FEC1::AB:2E0:7DFF:FEAB:C944
*Mar         1     01:25:45.655:     ICMPv6-ND:     PROBE     ->     REACH:
FEC1::AB:2E0:7DFF:FEAB:C944
*Mar         1     01:25:50.203:     ICMPv6-ND:     DELAY     ->     PROBE:
FE80::2E0:7DFF:FEAB:C944
*Mar         1     01:25:50.203:     ICMPv6-ND:     Sending     NS     for
FE80::2E0:7DFF:FEAB:C944 on FastEthernet0
*Mar         1     01:25:50.203:     ICMPv6-ND:     Received     NA     for
FE80::2E0:7DFF:FEAB:C944          on          FastEthernet0          from
FE80::2E0:7DFF:FEAB:C944
*Mar         1     01:25:50.203:     ICMPv6-ND:     PROBE     ->     REACH:
FE80::2E0:7DFF:FEAB:C944
```

- Descubrimiento de Vecinos en REDESPE

```
redespe#debug ipv6 nd
ICMP Neighbor Discovery events debugging is on
redespe#
01:10:10: ICMPv6-ND: STALE -> REACH: ::192.188.1.1
01:10:10: ICMPv6-ND: STALE -> DELAY: FEC0::CD:202:3FFF:FE8B:652A
01:10:15:  ICMPv6-ND:  Received  NS  for  FE80::2E0:B0FF:FE5A:9E8F  on
Ethernet0 from FE80::202:3FFF:FE8B:652A
01:10:15:  ICMPv6-ND:  Sending  NA  for  FE80::2E0:B0FF:FE5A:9E8F  on
Ethernet0
01:10:15: ICMPv6-ND: STALE -> DELAY: FE80::202:3FFF:FE8B:652A
01:10:15: ICMPv6-ND: DELAY -> PROBE: FEC0::CD:202:3FFF:FE8B:652A
01:10:15:  ICMPv6-ND:  Sending  NS  for  FEC0::CD:202:3FFF:FE8B:652A  on
Ethernet0
01:10:15:  ICMPv6-ND:  Received  NA  for  FEC0::CD:202:3FFF:FE8B:652A  on
Ethernet0 from FEC0::CD:202:3FFF:FE8B:652A
01:10:15: ICMPv6-ND: PROBE -> REACH: FEC0::CD:202:3FFF:FE8B:652A
01:10:20: ICMPv6-ND: DELAY -> PROBE: FE80::202:3FFF:FE8B:652A
```

```
01:10:20: ICMPv6-ND: Sending NS for FE80::202:3FFF:FE8B:652A on
Ethernet0
01:10:20: ICMPv6-ND: Received NA for FE80::202:3FFF:FE8B:652A on
Ethernet0 from FE80::202:3FFF:FE8B:652A
01:10:20: ICMPv6-ND: PROBE -> REACH: FE80::202:3FFF:FE8B:652A
```

4.2.9 Resumen

Para la Red 1 las direcciones IP quedan como se muestra en la siguiente tabla:

Tabla 4.1: Direcciones IP de los equipos de la Red 1

Equipo	**Dirección IP**	
PC	IPv4	10.10.1.12
	IPv6	fec1::ab:211:25ff:fe67:5a57
REDFISI	Ethernet0	10.10.1.1
	Serial 0	192.188.1.1

Para la Red 2 las direcciones IP quedan como se muestra en la siguiente tabla:

Tabla 4.2: Direcciones IP de los equipos de la Red 2

Equipo	**Dirección IP**	
Laptop	IPv4	10.10.1.11
	IPv6	fec0::cd:202:2fff:fe8b:652a
REDESPE	Ethernet	10.10.1.2
	Serial 0	65.172.180.65

Para la red que simula el Internet las direcciones IP quedan como se muestra en la siguiente tabla:

Tabla 4.3: Direcciones IP del router Internet

Equipo	Dirección IP	
INTERNET	Serial 0	65.172.180.66
	Serial 1	192.188.1.2

4.2.10 Conclusiones de la práctica

4.2.10.1 Después de realizar la configuración de los routers se pudo establecer un túnel manual.

4.2.10.2 Después de realizar la configuración de los routers se pudo establecer un túnel GRE.

4.2.10.3 Después de realizar la configuración de los routers se pudo establecer un túnel automático.

4.2.10.4 Fueron realizadas las respectivas pruebas de funcionamiento para cada uno de los tres túneles configurados, tales como ping y tracert.

4.3 Conexión de redes IPv4 con redes públicas IPv6

4.3.1 Finalidad de la práctica

4.3.1.1 Establecer un túnel entre la red IPv6 de netXperts Consulting y el Broker de Hexago, que permita acceder a sitios implementados en IPv6 a todos los equipos conectados a la red local.

4.3.2 Equipos utilizados

4.3.2.1 1 router Cisco 2500 con soporte dual stack

4.3.2.2 1 PC con sistema operativo Linux

4.3.2.3 1 PC con sistema operativo Windows

4.3.3 Descripción de la red

La red utilizada para la conexión a una red pública IPv6 está conformada por un router Cisco 2500 dual stack, el mismo que tiene la capacidad de poder conectar directamente los computadores de una red sin necesidad de tener un concentrador (hub, switch). En la red local se tiene un equipo con Windows XP Professional y otro con Fedora 2 Core, cada uno de los cuales cuenta con una dirección IPv4. Cabe recalcar que el router dual stack tiene asignada una IP pública a fin de establecer la comunicación con el servidor de túneles.

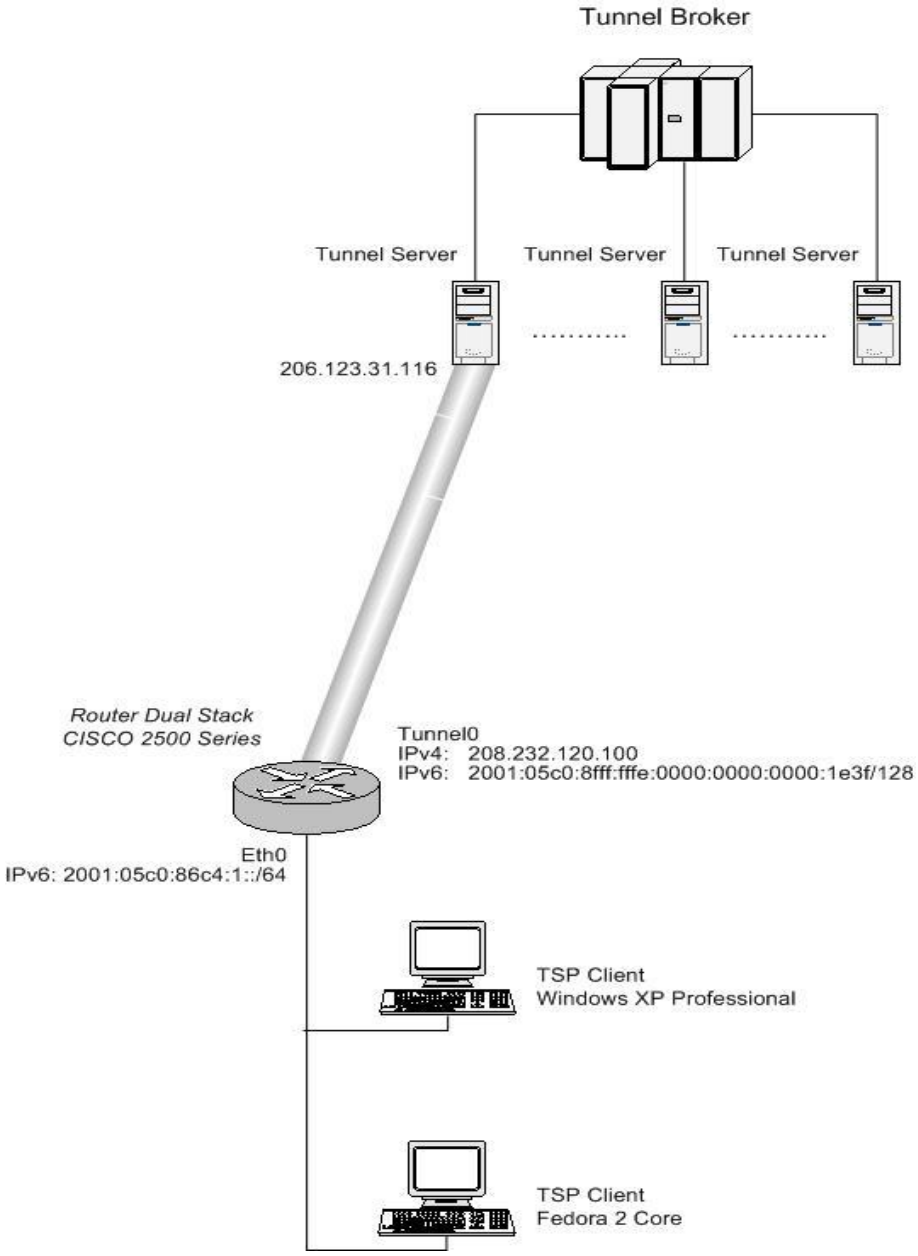


Figura 4.18: Descripción de la red

4.3.4 Implementación de Tunneling Público

El primer paso consiste en la creación de una cuenta de usuario en Freenet6, para lo cual es necesario ingresar la siguiente información:

- Nombre completo
- Nombre de cuenta de usuario

- Nombre de la empresa
- Dirección de la empresa
- Dirección de correo electrónico
- Número telefónico
- Como aprendió acerca de FreeNet
- Nivel de interés en IPv6
- Estado de desarrollo en IPv6
- Planes futuros por la experimentación en IPv6

Luego del proceso de registro se genera una clave que es enviada a la dirección de correo electrónico suministrada por el usuario.

El segundo paso es la descarga del software del Cliente TSP[41], el cual va a ser instalado en el equipo desde el cual se va a establecer el túnel. Existe un instalador del cliente TSP para cada uno de los siguientes sistemas operativos: Microsoft Windows XP, Microsoft Windows 2000, FreeBSD, NetBSD, Linux, Linux RPM, Linux Debian, Solaris 8, Solaris 9, OpenBSD, y MacOS X Darwin. Para el caso de estudio se descargó la versión para Linux.

El tercer y último paso es instalar el cliente TSP en el computador desde la cual se va a establecer el túnel. Para ello se siguen los siguientes pasos:

1. Iniciar sesión en Linux con el usuario root.
2. Descomprimir el archivo del cliente TSP en el directorio /home

[41] TSP: Tunnel Setup Protocol (Protocolo de Configuración de Túneles) Protocolo de señalización que sirve para configurar parámetros los extremos de un túnel.

3. Abrir una ventana de terminal, acceder al directorio tspc2 y ejecutar el siguiente comando:

make install target=linux installdir=/usr/local/tspc

```
[root@tesis tspc2]# make install target=linux installdir=/usr/local/tspc
mkdir -p bin
mkdir -p objs
make[1]: Entering directory `/home/crisale/Datos/tesis/tspc2/src/net'
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c net.c -o
../../objs/n et.o -Dlinux
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c net_rudp.c
-o ../../o bjs/net_rudp.o -Dlinux
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c net_tcp.c -
o ../../ob js/net_tcp.o -Dlinux
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c net_udp.c -
o ../../ob js/net_udp.o -Dlinux
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c net_ka.c -o
../../obj s/net_ka.o -Dlinux
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c net_cksm.c
-o ../../o bjs/net_cksm.o -Dlinux
make[1]: Leaving directory `/home/crisale/Datos/tesis/tspc2/src/net'
make[1]: Entering directory `/home/crisale/Datos/tesis/tspc2/src/lib'
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c base64.c -o
../../obj s/base64.o -Dlinux
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c cli.c -o
../../objs/c li.o -Dlinux
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c config.c -o
../../obj s/config.o -Dlinux
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c lib.c -o
../../objs/l ib.o -Dlinux
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c log.c -o
../../objs/l og.o -Dlinux
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c md5c.c -o
../../objs/ md5c.o -Dlinux
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c cnfchk.c -o
../../obj s/cnfchk.o -Dlinux
make[1]: Leaving directory `/home/crisale/Datos/tesis/tspc2/src/lib'
make[1]: Entering directory `/home/crisale/Datos/tesis/tspc2/src/tsp'
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c tsp_auth.c
-o ../../o bjs/tsp_auth.o -Dlinux
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c tsp_cap.c -
o ../../ob js/tsp_cap.o -Dlinux
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c
tsp_client.c -o ../.. /objs/tsp_client.o -Dlinux
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c tsp_net.c -
o ../../ob js/tsp_net.o -Dlinux
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c tsp_setup.c
-o ../../ objs/tsp_setup.o -Dlinux
make[1]: Leaving directory `/home/crisale/Datos/tesis/tspc2/src/tsp'
make[1]: Entering directory `/home/crisale/Datos/tesis/tspc2/src/xml'
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c xmlparse.c
-o ../../o bjs/xmlparse.o -Dlinux
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c xml_req.c -
o ../../ob js/xml_req.o -Dlinux
gcc -O2 -g -Wall -I../../include -I../../platform/linux -c xml_tun.c -
o ../../ob js/xml_tun.o -Dlinux
make[1]: Leaving directory `/home/crisale/Datos/tesis/tspc2/src/xml'
```

```
make[1]: Entering directory
`/home/crisale/Datos/tesis/tspc2/platform/linux'
gcc -g -Wall -I../../include -I../../platform/linux -c tsp_local.c -o
../../objs /tsp_local.o -Dlinux
gcc -g -Wall -I../../include -I../../platform/linux -c tsp_tun.c -o
../../objs/t sp_tun.o -Dlinux
gcc -g -Wall -I../../include -I../../platform/linux -o ../../bin/tspc
../../objs /*.o
make[1]: Leaving directory
`/home/crisale/Datos/tesis/tspc2/platform/linux'
make[1]: Entering directory `/home/crisale/Datos/tesis/tspc2/template'
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/home/crisale/Datos/tesis/tspc2/template'
make[1]: Entering directory `/home/crisale/Datos/tesis/tspc2/conf'
Generating basic configuration file
chmod 700 ../bin/tspc.conf.sample
make[1]: Leaving directory `/home/crisale/Datos/tesis/tspc2/conf'
make[1]: Entering directory `/home/crisale/Datos/tesis/tspc2/man'
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/home/crisale/Datos/tesis/tspc2/man'
mkdir -p /usr/local/tspc
make[1]: Entering directory `/home/crisale/Datos/tesis/tspc2/src/net'
make[1]: Nothing to be done for `install'.
make[1]: Leaving directory `/home/crisale/Datos/tesis/tspc2/src/net'
make[1]: Entering directory `/home/crisale/Datos/tesis/tspc2/src/lib'
make[1]: Nothing to be done for `install'.
make[1]: Leaving directory `/home/crisale/Datos/tesis/tspc2/src/lib'
make[1]: Entering directory `/home/crisale/Datos/tesis/tspc2/src/tsp'
make[1]: Nothing to be done for `install'.
make[1]: Leaving directory `/home/crisale/Datos/tesis/tspc2/src/tsp'
make[1]: Entering directory `/home/crisale/Datos/tesis/tspc2/src/xml'
make[1]: Nothing to be done for `install'.
make[1]: Leaving directory `/home/crisale/Datos/tesis/tspc2/src/xml'
make[1]: Entering directory
`/home/crisale/Datos/tesis/tspc2/platform/linux'
make[1]: Nothing to be done for `install'.
make[1]: Leaving directory
`/home/crisale/Datos/tesis/tspc2/platform/linux'
make[1]: Entering directory `/home/crisale/Datos/tesis/tspc2/template'
Installing templates
make[1]: Leaving directory `/home/crisale/Datos/tesis/tspc2/template'
make[1]: Entering directory `/home/crisale/Datos/tesis/tspc2/conf'
Generating basic configuration file
chmod 700 ../bin/tspc.conf.sample
make[1]: Leaving directory `/home/crisale/Datos/tesis/tspc2/conf'
make[1]: Entering directory `/home/crisale/Datos/tesis/tspc2/man'
Installing man pages
mkdir -p /usr/local/tspc/man/man5
mkdir -p /usr/local/tspc/man/man8
cp man5/tspc.conf.5 /usr/local/tspc/man/man5
cp man8/tspc.8 /usr/local/tspc/man/man8
To view man pages run :
man -M /usr/local/tspc/man tspc
man -M /usr/local/tspc/man tspc.conf
make[1]: Leaving directory `/home/crisale/Datos/tesis/tspc2/man'
```

4. Acceder al directorio /usr/local/tscp/bin y modificar el archivo tspc.conf con la información de la cuenta de usuario, de la siguiente manera:

userid=jpaobe1904

passwd=8LSiiVioK!

server=broker.freenet6.net

5. Una vez modificada esta información, desde una ventana de terminal se ejecuta el archivo tspc.conf de la siguiente forma:

./tspc -vvv

```
[root@tesis bin]# ./tspc -vvv
tspc - Tunnel Setup Protocol Client v2.1.1
Initializing (use -h for help)

 Connecting to server with reliable udp
Using TSP protocol version 2.0.0
Establishing connection with tunnel broker...
Getting capabilities from server
Connection established
Authenticating jpaobe1904
Using authentification mecanism DIGEST-MD5
Authentication success
Asking for a tunnel
sent: Content-length: 201
<tunnel action="create" type="v6anyv4" proxy="no">
 <client>
  <address type="ipv4">10.10.0.55</address>
<keepalive  interval="30"><address  type="ipv6">::</address></keepalive>
</client>
</tunnel>

recv:
200 Success
<tunnel action="info" type="v6udpv4" lifetime="604800">
  <server>
    <address type="ipv4">206.123.31.116</address>
    <address
type="ipv6">2001:05c0:8fff:fffe:0000:0000:0000:1e16</address>
  </server>
  <client>
    <address type="ipv4">200.107.11.35</address>
    <address
type="ipv6">2001:05c0:8fff:fffe:0000:0000:0000:1e17</address>
    <keepalive interval="30">
      <address
type="ipv6">2001:05c0:8fff:fffe:0000:0000:0000:1e16</address>
    </keepalive>
  </client>
</tunnel>

 Processing response from server
sent: Content-length: 35
```

```
<tunnel action="accept"></tunnel>

Got tunnel parameters from server, setting up local tunnel
Checking for linux ipv6 support...
IPv6 support found
keepalive interval: 30

Going daemon, check tspc.log for tunnel creation status
[root@tesis bin]#
```

6. Para cerciorarse de que el protocolo IPv6 está ya en funcionamiento, se observa la configuración de la tarjeta de red

```
[root@tesis /]# sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:10:4B:C5:1E:D0
          inet addr:10.10.0.55  Bcast:10.10.0.63  Mask:255.255.255.240
          inet6 addr: fe80::210:4bff:fec5:1ed0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:540300 errors:0 dropped:0 overruns:0 frame:0
          TX packets:372636 errors:0 dropped:0 overruns:0 carrier:48
          collisions:23 txqueuelen:1000
          RX bytes:236290770 (225.3 Mb)  TX bytes:27979005 (26.6 Mb)
          Interrupt:217 Base address:0xbc00

eth0:1    Link encap:Ethernet  HWaddr 00:10:4B:C5:1E:D0
          inet      addr:166.20.240.105           Bcast:166.20.240.255
Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3967 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4141 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1487123 (1.4 Mb)  TX bytes:410780 (401.1 Kb)
          Interrupt:217 Base address:0xbc00

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:16937 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16937 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1476652 (1.4 Mb)  TX bytes:1476652 (1.4 Mb)

tun       Link encap:Point-to-Point Protocol
          inet6 addr: 2001:5c0:8fff:fffe::1e17/128 Scope:Global
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1280  Metric:1
          RX packets:3967 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4141 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:1487123 (1.4 Mb)  TX bytes:410780 (401.1 Kb)
```

7. Para comprobar la funcionalidad del túnel se puede utilizar un navegador de Internet y acceder a la página web de HEXAGO (www.hexago.com). Si el túnel está trabajando de forma correcta se mostrará en la parte superior izquierda de la página, la dirección IPv6 que fue asignada para este procedimiento.

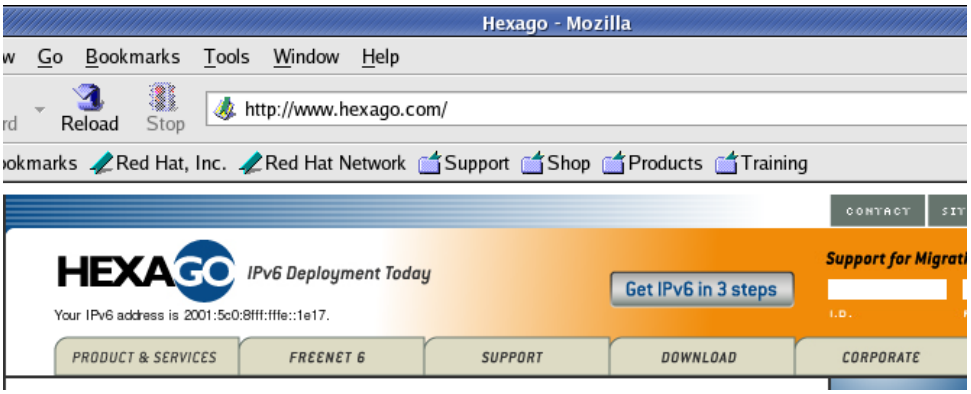


Figura 4.19: Página Web de HEXAGO

En el sitio de Información IPv6 (www.ipv6.org) también se puede observar la dirección IPv6 proporcionada al seguir dicho procedimiento.

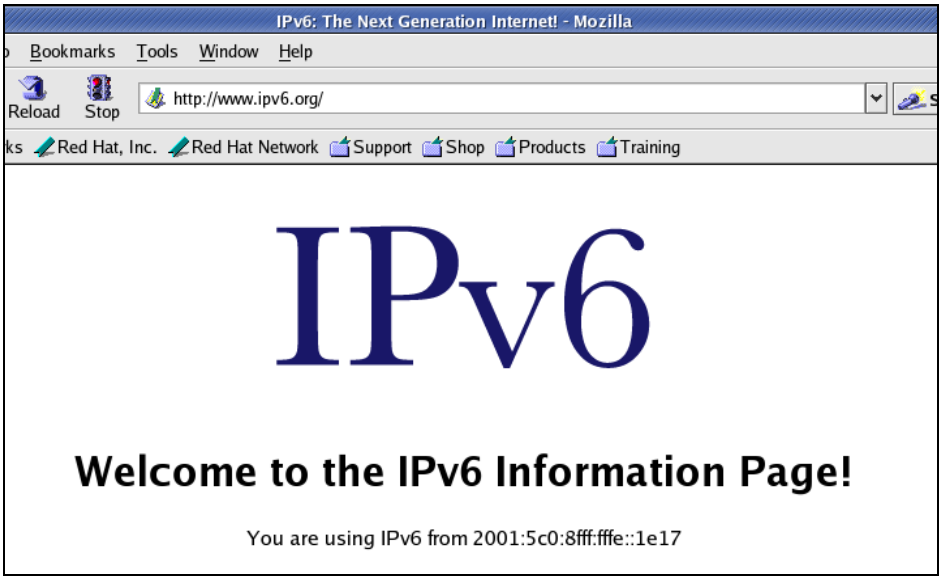


Figura 4.20: Página web de IPv6

En la página del proyecto KAME www.kame.net se puede observar una tortuga moviéndose cuando se tiene habilitado el protocolo IPv6, además en la parte inferior de la misma se muestra la dirección IPv6 asignada.

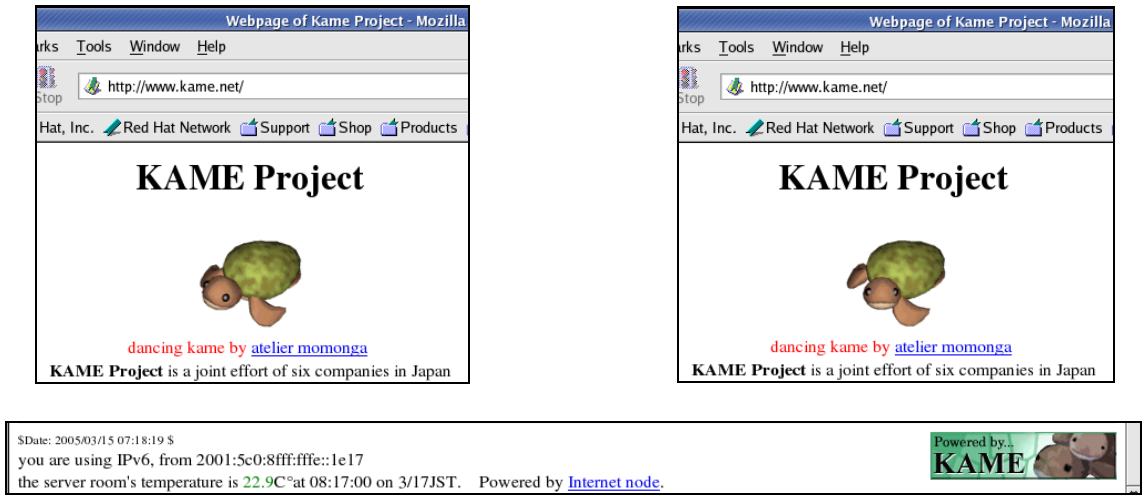


Figura 4.21: Página web del proyecto KAME

Adicionalmente se realiza un tracert hacia el sitio de KAME para ver la ruta que toma el paquete hasta llegar a su destino y una vez más comprobar la funcionalidad del túnel.

```
[root@tesis /]# traceroute6 www.kame.net
traceroute to www.kame.net (2001:200:0:8002:203:47ff:fea5:3085) from
2001:5c0:8fff:fffe::1e17, 30 hops max, 16 byte packets
 1   2001:5c0:8fff:fffe::1e16 (2001:5c0:8fff:fffe::1e16)   398.213 ms
567.911 ms  487.187 ms
 2    ipv6.hexago.com  (2001:5c0:0:2::18)    395.018  ms    536.563  ms
616.578 ms
 3   sl-bb1v6-nyc-t-23.sprintv6.net (3ffe:2900:2001:5::1)   550.538 ms
682.656 ms  630.096 ms
 4   sl-bb1v6-rly-t-1003.sprintv6.net (2001:440:1239:100a::2)   705.434
ms  584.423 ms  414.928 ms
 5    3ffe:2900:b:e::2  (3ffe:2900:b:e::2)    610.346  ms    566.598  ms
607.937 ms
 6  plt001ix06.IIJ.Net (2001:240:bb62:8000::4001)  628.084 ms  748.551
ms  668.134 ms
 7   otm6-bb1.iij.net (2001:240:bb20:f001::2)   709.133 ms   766.473 ms
otm6-bb0.iij.net (2001:240:bb20:f000::1)  709.766 ms
 8   tky001ix06.iij.net (2001:240:100:1::30)   700.496 ms   707.798 ms
otm6-gate0.iij.net (2001:240:100::204)  750.891 ms
 9  hitachi1.otemachi.wide.ad.jp (2001:200:0:1800::9c4:2)   630.516 ms
665.009 ms pc6.otemachi.wide.ad.jp (2001:200:0:1800::9c4:0)  1149 ms
10   hitachi1.otemachi.wide.ad.jp (2001:200:0:1802:240:66ff:fe10:cf7c)
615.87       ms              661.928       ms       pc3.yagami.wide.ad.jp
```

```
(2001:200:0:1c04::1000:2000)  646.452 ms
11     gr2000.k2.wide.ad.jp   (2001:200:0:4819::2000:1)     596.534   ms
592.329 ms  582.195 ms
12     gr2000.k2.wide.ad.jp   (2001:200:0:4819::2000:1)     625.681   ms
orange.kame.net   (2001:200:0:8002:203:47ff:fea5:3085)     608.953   ms
609.467 ms
```

Finalmente, para configurar el router CISCO que permitirá acceder al túnel a todos los clientes de la red IPv6, se debe realizar una configuración adicional siguiendo los siguientes pasos:

1. Modificar el archivo tspc.conf con datos adicionales que permitirán la generación de un archivo TXT, el mismo que contendrá la configuración del router. Para ello se debe proporcionar la dirección IPv4 real del router. Los datos a ser modificados se muestran a continuación:

   host_type=router

   prefixlen=48

   if_tunnel_v6v4=Tunnel0

   if_prefix=e0/0

   proxy_client=yes

   client_v4=208.232.120.100

   keepalive=no

   template=cisco

2. Una vez modificada esta información, desde una ventana de terminal se ejecuta el archivo tspc.conf de la siguiente forma:

   ./tspc -vvv

```
[root@tesis bin]# ./tspc -vvv
tspc - Tunnel Setup Protocol Client v2.1
Initializing (use -h for help)
```

```
if_tunnel_v6udpv4 not specified, forcing tunnel_mode to v6v4.

Connecting to server with reliable udp
Using TSP protocol version 2.0.0
Getting capabilities from server
Authenticating cemnetx
Authentication success
Asking for a tunnel
Processing response from server
Got tunnel parameters from server, setting up local tunnel
tspSetupInterface beginning
sáb 05/03/2005
17:03
IPv4 tunnel server address configured : 206.123.31.116
cisco Router will be configured as IPv6 router and it will do router
advertisements for autoconfiguration
Configuring IPv6_forwading on network interface
Script completed sucessfully
Your IPv6 address is 2001:05c0:8fff:fffe:0000:0000:0000:1e3f
TSP session done
```

3. Abrir el archivo TXT generado, CISCOIPv6.txt y copiar la información ahí descrita en la configuración del router

```
! Add these lines to your Cisco configuration
! Script launched from a Microsoft environment
!
ipv6 unicast-routing
!
interface Tunnel0
 ipv6 address 2001:05c0:8fff:fffe:0000:0000:0000:1e3f/128
 tunnel source 208.232.120.100
 tunnel destination 206.123.31.116
 tunnel mode ipv6ip
!
ipv6 route ::/0 Tunnel0
!
interface ethernet0
 ipv6 address 2001:05c0:86c4:1::/64 eui-64
 ipv6 nd prefix-advertisement 2001:05c0:86c4:1::/64 43200 43200 onlink
autoconfig
```

4. Conectar una PC al router, ver la configuración del mismo y observar como el router asigna el prefijo con el cual se va a conectar a sitios IPv6, gracias a la funcionalidad del protocolo Neighbor Discovery.

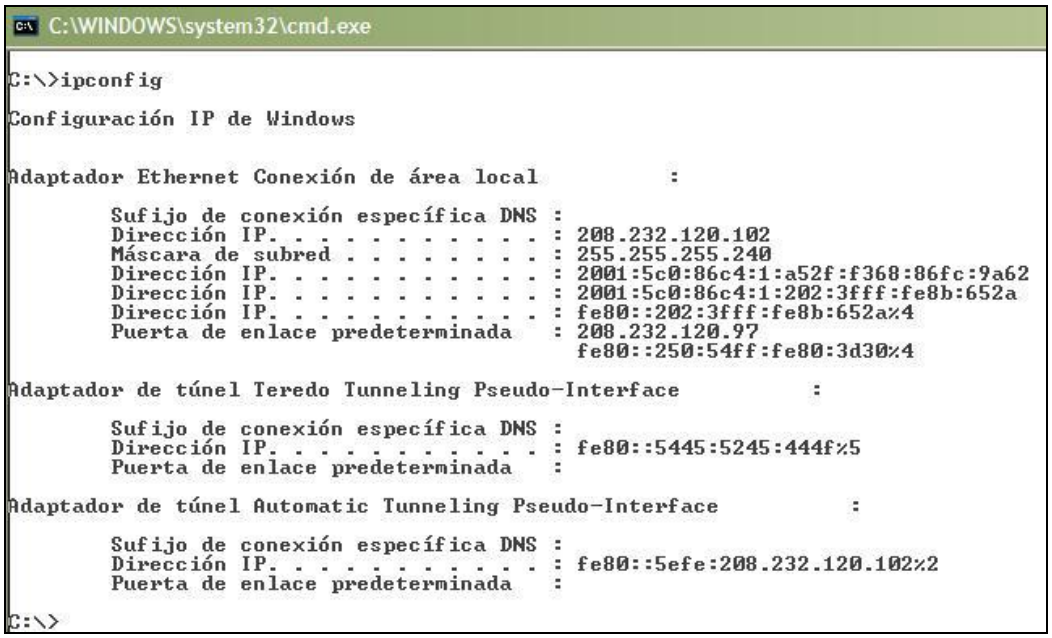


Figura 4.22: Muestra la configuración de la tarjeta de red

5. Realizar pruebas de ping desde el router hacia la máquina conectada al mismo

```
acuenca#ping ipv6 2001:5c0:86c4:1:202:3fff:fe8b:652a

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to
2001:5C0:86C4:1:202:3FFF:FE8B:652A, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms
```

6. Desde la máquina conectada al router, se ingresa a distintos sitos IPv6 para que se despliegue la dirección IPv6 que fue asignada por el router.

Desde el sitio de HEXAGO (www.hexago.com) se puede observar en la parte superior izquierda la dirección IPv6 con la cual se está conectando al mismo.

Figura 4.23: Página web de HEXAGO

De igual forma desde la página de información IPv6 (www.ipv6.org) se muestra la dirección IPv6 con la cual se establece la conexión a dicho sitio.

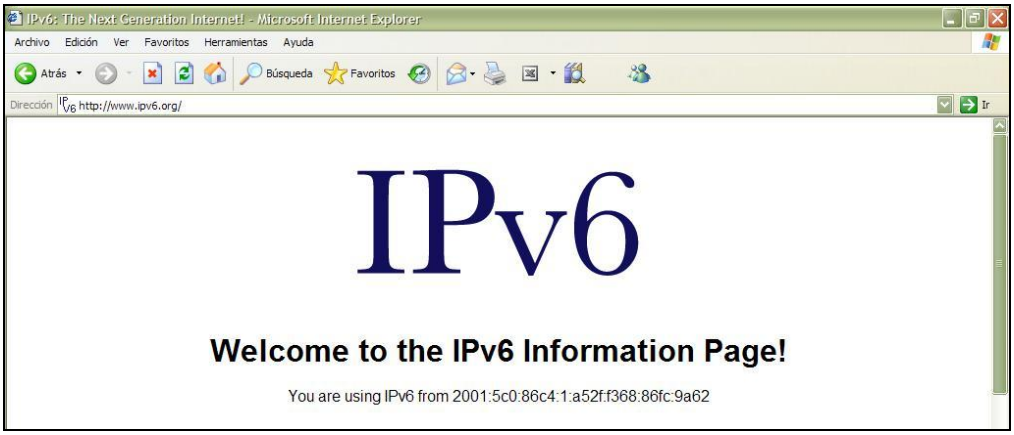


Figura 4.24: Página web de IPv6

Adicionalmente se ingresa al sitio del proyecto KAME (www.kame.net) para observar la tortuga en movimiento con lo cual se comprueba que estamos accediendo con una dirección IPv6 a dicho sitio. En la parte inferior de esta página también se muestra la dirección IPv6.

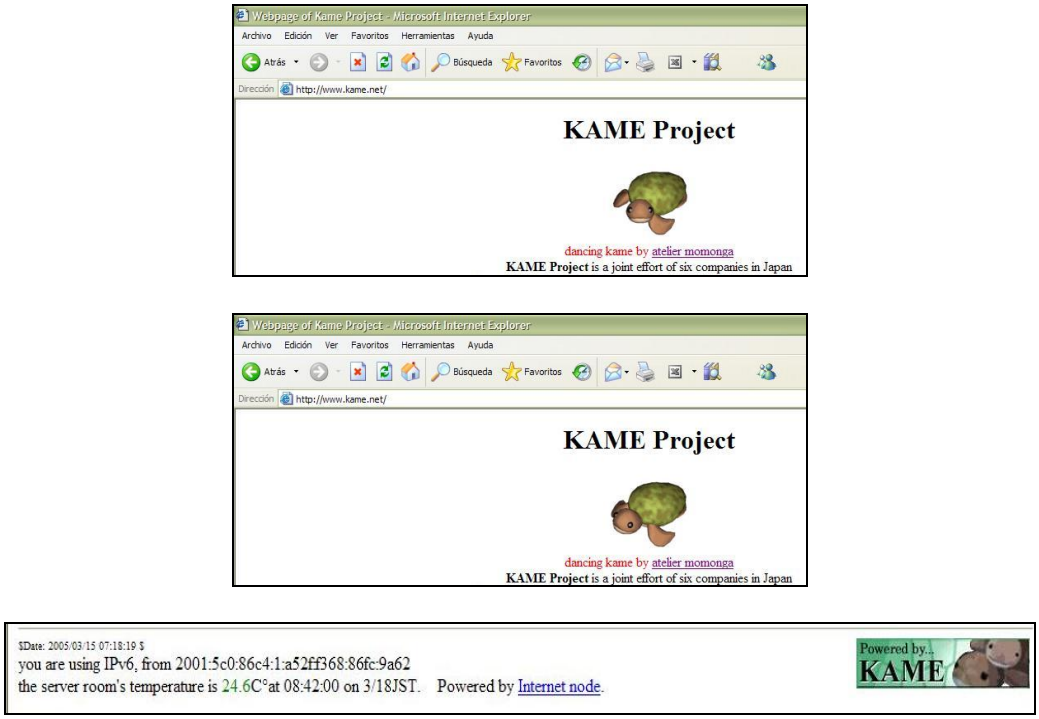


Figura 4.25: Página web del proyecto KAME

4.3.5 Conclusiones de la práctica

4.3.5.1 Se estableció un túnel entre la red IPv6 de netXperts Consulting y el Broker de Hexago, el cual permitió el acceso a sitios implementados en IPv6 a todos los equipos conectados a la red de la empresa, como por ejemplo el sitios de HEXAGO, del proyecto KAME y de la organización IPv6.

CAPITULO V

5 CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- Las principales ventajas que el protocolo IPv6 tiene en comparación con IPv4 se refieren al espacio de direccionamiento, seguridad, movilidad, calidad y clases de servicio. Por tales razones el protocolo IPv6 está llamado a ser el estándar de la nueva generación de Internet.
- La transición entre los protocolos IPv4 e IPv6 es un proceso en desarrollo a nivel mundial y por ello los mecanismos de coexistencia entre dichas redes adquieren mayor importancia, pues se constituyen en una solución a corto y mediano plazo. Dentro de dichos mecanismos se destaca al tunneling entre redes como la alternativa más factible de implementar para acceder a redes y servicios IPv6.
- Una vez finalizado el presente proyecto se tesis se logró implementar tunneling entre islas IPv6 a través de la red IPv4 existente, luego de realizar las configuraciones necesarias en la infraestructura de comunicaciones de la empresa “netXperts Consulting S.A.”. De igual manera se implementó una solución de tunneling para brindar acceso a redes públicas IPv6 desde la red privada de la empresa.
- En lo referente al estudio de los fundamentos teóricos del protocolo IPv6 cabe recalcar que el Internet se constituyó en la mayor fuente de información, pues la bibliografía referente al tema es bastante limitada y solamente cubre aspectos básicos.
- Luego de la implementación del tunneling se realizaron diversas pruebas de funcionalidad, en las cuales se pudo verificar la comunicación entre redes IPv6 y la comunicación con redes públicas IPv6.
- El principal limitante encontrado durante el desarrollo del presente proyecto de tesis se refiere a la poca disponibilidad de equipos de comunicaciones con soporte para IPv6, los mismos que eran necesarios para realizar las pruebas de funcionalidad del tunneling desde la red IPv6 de la empresa “netXperts Consulting S.A.”

- En nuestro país la tecnología IPv6 no tiene el nivel de desarrollo que se puede observar a nivel mundial. Esto se debe en gran parte a que no existen muchas instituciones interesadas en emprender proyectos enfocados hacia el desarrollo de IPv6.

5.2 RECOMENDACIONES

- Debido a que los fundamentos teóricos del protocolo IPv6 abarcan una gran cantidad de conocimientos, es recomendable tomarse el tiempo necesario para hacer un estudio minucioso que permita tener un alto nivel de comprensión al momento de desarrollar soluciones para redes IPv6.
- Las entidades públicas y privadas deben tomar en cuenta de que en un futuro cercano, una mayor cantidad de recursos estarán disponibles sobre redes IPv6. Por tal razón se recomienda hacer una actualización de los sistemas y equipos de comunicaciones, a fin de tener soporte para IPv6 y aprovechar las ventajas que el mismo puede ofrecer.
- Antes de emprender algún proyecto de investigación referente a IPv6, es recomendable cerciorarse de que se cuenta con las facilidades para el acceso a infraestructura de red con soporte dual stack, que permita tener una mejor idea del funcionamiento del protocolo IPv6.
- Las instituciones educativas están llamadas a poner en marcha los procesos de investigación sobre los fundamentos teóricos y técnicos del protocolo IPv6. Por tal razón se recomienda que la Facultad de Sistemas de la ESPE incluya materias afines al tema en los pénsum de estudio. Además se recomienda la implementación de un laboratorio de IPv6, que cuente con toda la infraestructura de red necesaria para que los estudiantes puedan poner en práctica todos los conocimientos teóricos aprendidos.
- Es recomendable que la ESPE realice un análisis de factibilidad para convertirse en un proveedor de servicios IPv6, y de esta forma estar al mismo nivel de otras instituciones educativas del continente.

GLOSARIO

Autenticación: Mecanismo de seguridad por el que un sistema comprueba la identidad de una persona, por medio de la utilización de un nombre de usuario y una contraseña.

Backbone: Es la infraestructura de conexión principal de una red y está constituida por los enlaces de mayor velocidad dentro de dicha red.

Broadcast: Técnica que consiste en mandar el mismo mensaje simultáneamente a múltiples receptores dentro de la misma red.

Bootstrap: Programa utilizado para permitir el arranque de un dispositivo.

Broker: Entidad que actúa como intermediario entre un alguien que necesita un servicio y alguien que ofrece dicho servicio. Actúa como agente y no toma ninguna posición propia durante la negociación.

Buffer: Espacio de memoria dedicado a almacenar temporalmente la información que debe procesar un dispositivo, de tal forma que no afecte al rendimiento del mismo.

Concentrador: Es un dispositivo de red que conecta múltiples computadoras para conformar una red de datos.

Conmutación: Proceso por el que los paquetes son recibidos, almacenados y transmitidos al puerto de destino apropiado.

Datagrama: Unidad de información transmitida por los protocolos de nivel de red. El datagrama contiene no sólo los datos: entre otras informaciones, se añade la dirección del emisor de los datos, así como la de su destinatario. Esta información le permite al protocolo TCP/IP encaminar (transportar a través de las redes y host

que hagan falta) los datos desde el origen a su destino. Los datagramas, al igual que los mensajes de correo electrónico, poseen su propio encabezado.

Dominio: Conjunto de ordenadores que comparten una característica común, como el estar en el mismo país, en la misma organización o en el mismo departamento.

Dual Stack: Denominación que se da a los equipos que tienen la capacidad de manejar las pilas de protocolos IPv4 e IPv6.

Encriptación: Tratamiento de un conjunto de datos, contenidos o no en un paquete, a fin de impedir que nadie excepto el destinatario de los mismos pueda leerlos. Hay muchos tipos de cifrado de datos, que constituyen la base de la seguridad de la red.

Gateway: Puerta de enlace. Computador que permite el enlace entre dos redes, enrutando datagramas IP y convirtiendo protocolos o mensajes desde una red a otra.

GRE: (Generic Route Encapsulation) Protocolo de tunneling desarrollado por CISCO, que encapsula una amplia variedad de paquetes de protocolos dentro de tuneles IP, creando un enlace virtual punto a punto hacia routers Cisco desde sitios remotos, sobre una infraestructura IP.

Host: Computador que pertenece a una red y que puede ser configurado para desarrollar tareas específicas.

Latencia: Tiempo de espera hasta la recepción de un dato que se ha solicitado a un dispositivo. Es calculado en nanosegundos (ns) y milisegundos (ms).

Log: Archivo que registra movimientos y actividades de un determinado programa o dispositivo. Es utilizado como mecanismo de control y estadística.

Loopback: Dirección asignada por defecto a todo computador por parte del sistema operativo, la cual permite realizar pruebas de configuración de ciertos servicios.

Nodo: Dispositivo de comunicación que interviene en una red, ya sea ejecutando labores de interconexión ó simplemente formando parte de la misma.

Ping: (Packet INternet Groper – Buscador de paquetes de Internet). Es un comando que se utiliza para saber si un computador determinado se encuentra conectado a la red y está encendido. Para utilizar este comando se debe especificar una dirección IP o un nombre de equipo, al cual se desea rastrear.

Protocolo: Se le llama protocolo de red o protocolo de comunicación al conjunto de reglas que controlan la secuencia de mensajes que ocurren durante una comunicación entre entidades que forman una red. En este contexto, las entidades son programas o dispositivos de comunicaciones capaces de interactuar en una red.

Ruteo: Más conocido como encaminamiento, es el mecanismo por el que en una red los paquetes de información se hacen llegar desde su origen a su destino final, siguiendo un camino o ruta a través de la red. En una red grande o en un conjunto de redes interconectadas el camino a seguir hasta llegar al destino final puede suponer transitar por muchos nodos intermedios.

Tracert: Es un comando que se usa para seguir la ruta o camino que se toma en una red, utilizando el protocolo TCP/IP.

Trama: Agrupación lógica de información enviada como unidad de datos en un medio de transmisión.

Topología: Disposición física de los nodos de una red.

BIBLIOGRAFÍA

El modelo OSI y TCP/IP

- http://www.monografias.com/trabajos13/modosi/modosi.shtml
- http://www.zator.com/Hardware/H12_2.htm
- http://web.frm.utn.edu.ar/comunicaciones/modelo_osi.html
- http://mx.geocities.com/lemt78/
- http://usuarios.lycos.es/janjo/janjo1.html
- http://eia.udg.es/~atm/tcp-ip/tema_4_2_1.htm
- http://ditec.um.es/laso/docs/tut-tcpip/3376fm.html

Router

- http://www.hack-box.info/libro/routers.html
- http://www.cisco.com/en/US/products/hw/routers/ps233/products_installation_and_configuration_guide_chapter09186a008007e271.html#30327
- http://www.verio.com/support/view_article.cfm?doc_id=463
- http://www.htmlweb.net/redes/routers/routers_2.html
- http://www.cisco.com/en/US/products/sw/iosswrel/ps5187/products_command_reference_chapter09186a00801d6610.html#wp1818356
- http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122sup/122csum/csum1/122csip2/1sfbgp1.htm#1074338

IPv6

- http://www.rau.edu.uy/ipv6/
- http://neutron.ing.ucv.ve/revista-e/No8/MEPadilla%5CArticulo%20IPv6.htm

- http://fmc.axarnet.es/tcp_ip/tema-03/tema-03.htm
- http://imasd.elmundo.es/imasd/ipv6/queesipv6.html
- http://internetng.dit.upm.es/ponencias-jing/2002/fernandez/Evolucion-IPv4-IPv6-David-Fernandez.PDF
- http://www.6sos.org/
- http://www.microsoft.com/windowsserver2003/technologies/ipv6/default.mspx

RFCs

- http://www.ietf.org/rfc.html

RFC 3056	Connection of IPv6 Domains via IPv4 Clouds
RFC 2893	Transition Mechanisms for IPv6 Hosts and Routers
RFC 791	Internet Protocol
RFC 3587	IPv6 Global Unicast Address Format
RFC 2374	An IPv6 Aggregatable Global Unicast Address Format
RFC 3513	IPv6 Addressing Architecture