UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERÍA

GUÍA DE IMPLEMENTACIÓN DE REDES PRIVADAS VIRTUALES (VPN), FIREWALLS Y CALIDAD DE SERVICIO (QoS) CON IPV6.

PARA OPTAR AL GRADO DE:

INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTADO POR:

KARLA GERLADINA IGLESIAS PORTAL

ASESOR:

ING. CARLOS LÓPEZ LINARES

AGOSTO DE 2007

EL SALVADOR, CENTROAMÉRICA

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERÍA
ESCUELA DE COMPUTACIÓN

RECTOR

ING. FEDERICO MIGUEL HUGUET RIVERA

VICERRECTOR

PADRE VICTOR BERMUÍDEZ

SECRETARIO

LIC. MARIO RAFAEL OLMOS

DECANO DE LA FACULTAD DE INGENIERÍA

ING. ERNESTO GODOFREDO GIRÓN

UNIVERSIDAD DON BOSCO

FACULTAD DE INGENIERÍA

ESCUELA DE COMPUTACIÓN

COMITÉ EVALUADOR DEL TRABAJO DE GRADUACIÓN

ING. CARLOS LÓPEZ LINARES

ASESOR

ING. RAFAEL CRISTOBAL HERNÁNDEZ

JURADO

ING. JOSÉ ALBERTO DÁVILA

JURADO

ING. EDGARDO ALBERTO ROMERO MASIS

JURADO

DEDICATORIA

Cada palabra escrita en este documento, cada hoja, cada alegría obtenida cuando las pruebas con ipv6 han funcionado de manera exitosa, cada momento se lo dedico a mis padres Gladys de Iglesias y Carlos Iglesias, de igual manera a mi hermano Juan Carlos Iglesias. Sus esfuerzos para que yo pudiera finalizar este proyecto han sido transcendentales, dignos de una familia con amor.

También dedico mi trabajo de graduación a mis primos, desde el mayor hasta el menor, a mi hermano Juan Carlos, que se encuentran luchando para obtener un título universitario, espero que tomen muchas fuerzas y ganas de luchar para lograrlo. No se cansen, deben de seguir adelante.

Dedico mi trabajo, de una forma especial a Avelino Mahinga Filho, de origen angoleño, por su apoyo, por brindarme a cada momento consejos, palabras de aliento, gracias por haberme escuchado todas mis angustias y brindarme una sonrisa para solucionar mis problemas y creer en mi esfuerzo.

Este trabajo esta dedicado para todas las personas que deseen investigar una nueva tecnología, y quieran ser vanguardistas. El esfuerzo es enorme cuando no se tiene mucho material de apoyo, o cuando no existe fuentes bibliográficas en nuestro idioma, y pues solo existen en otros lenguajes.

Para todas esas personas, que piensan que no se puede, por que el reto es grande o imposible, les dedico este trabajo, por que con esfuerzo, disciplina, confianza en Dios, se puede lograr hacerlo.

AGRADECIMIENTOS

En primer lugar agradezco a Dios todo poderoso, por que él es mi creador, ha sido siempre mi guía, mi luz, mi apoyo incondicional en mi vida. Sin Dios no hubiera alcanzado ésta meta que me propuse hace unos años. Gracias a su poder he finalizado mi tesis exitosamente. Agradezco de todo corazón a mis padres Gladys y Carlos, por todos los consejos que me han brindado y por llevarme de la mano durante el trayecto de mi vida, también a mi hermano Juan Carlos.

Por que el amor de mi familia me ha dado fuerzas para luchar contra toda adversidad, sus esperanzas y anhelos también son los míos, y el esfuerzo que he realizado en la elaboración de esta tesis, también es el esfuerzo de mis padres y mi hermano. Con todo mi corazón, gracias mamá por que sin usted no hubiera obtenido este logro, que ahora es nuestro, gracias por su infinito amor, por cuidarme y guiarme por el camino del bien, por enseñarme a leer, y formarme objetivos en mi vida, por estar conmigo en cada etapa buena o difícil.

Le doy gracias a mi tío Mardoqueo Iglesias y tía Luisa López por apoyarme emocionalmente y económicamente en todo este camino de mi estudio. Gracias tío Mardoqueo por ser mi segundo padre. También agradezco a mi familia materna, mis abuelos y tíos, por sus consejos, y dedicación hacia mi persona, nunca me han dejado sola, me han ayudado con sus oraciones para seguir adelante.

Agradezco de manera especial, a una persona que me enseñó a luchar por los objetivos trazados no importa distancias o sufrimientos, lo importante es lograr la meta. Esta persona es de origen africano, su nombre es Avelino Mahinga Filho, yo, lo admiro mucho por su valentía de lucha en la vida, por su dedicación y empeño, te agradezco por tu apoyo, tus consejos y tu amor.

Estoy muy agradecida con mis amigos, que juntos hemos luchado para lograr la meta al final, gracias por su ayuda a todos. Además agradezco a mis hermanos en Cristo Olga y familia Bran, por que gracias a sus oraciones pude tener fuerzas y cerrar un objetivo primordial en mi vida.

INDICE

INTRODUCCIÓN **1**

ANTECEDENTES **2**

IMPORTANCIA DE LA INVESTIGACIÓN. **4**

SITUACIÓN ACTUAL DE IPV6 EN EL SALVADOR. **6**

1. INTRODUCCIÓN A IPV6 ..... **9**

1.1 LIMITACIONES DE IPV4 9
1.2 CARACTERÍSTICAS DE IPV6 10
1.3 DIFERENCIAS ENTRE IPV4 E IPV6 12
1.4 TERMINOLOGÍA DE IPV6 14
1.5 BENEFICIOS E INNOVACIÓN 16
1.6 LA CABECERA DE IPV6 17
1.6.1 Estructura del paquete de IPv6 17
1.6.2 Estructura general de la cabecera 18
1.6.3 Los campos en la cabecera de IPv6 20
1.6.4 Cabeceras de extensión de IPv6 23
1.7 DIRECCIONAMIENTO EN IPV6 25
1.7.1 Direcciones Unicast 26
1.7.2 Direcciones Anycast IPv6 28
1.7.3 Direcciones multicast 29
1.7.4 Representación de las direcciones IPv6 30
1.7.4.1 Prefijos 32
1.7.4.2 Direcciones Reservadas 33
1.8 CONFIGURANDO IPV6 EN LOS SISTEMAS OPERATIVOS. 34

2. SEGURIDAD EN IPV6 ..... **36**

2.1 CARACTERÍSTICAS DE SEGURIDAD EN IPV6 37
2.2 ASPECTOS TÉCNICOS DE LA SEGURIDAD EN INTERNET 37
2.2.1. Diseño 39
2.2.2. Elementos básicos de seguridad 40
2.2.3. Políticas de seguridad 41
2.2.4. Amenazas y ataques en la red. 43
2.2.5 Tipos de ataques a redes 44
2.3 REQUERIMIENTOS BÁSICOS DE SEGURIDAD 48

2.3.1 Confidencialidad 48
2.3.2 Integridad 48
2.3.3 Autenticidad 49
2.3.4 No repudio 50
2.3.5 Encriptación 50
2.3.5.1 Clave privada (simétrica). 51
2.3.5.2 Clave pública (asimétrica). 51
2.3.5.3 Firma Digital. 53
2.4 SEGURIDAD EN INTERNET Y ALGUNAS SOLUCIONES 56
2.4.1 Soluciones Actuales. 57
2.4.2 Filtros de paquetes y Firewalls. 57
2.5 PROTECCIÓN EN LA CAPA DE TRANSPORTE. 58
2.6 APICACIONES DE SEGURIDAD. 59
2.7 DIFERENCIAS DE SEGURIDAD ENTRE IPV4 E IPV6 62

3. MECANISMOS DE SEGURIDAD 65

3.1 TERMINOLOGÍA DE LA SEGURIDAD EN IPV6 66
3.2 ASOCIACIONES DE SEGURIDAD 67
3.3 LA CABECERA DE AUTENTIFICACIÓN (AH) 68
3.3.1 Formato de la cabecera de Autentificación 69
3.3.2 Localización de la Cabecera de Autentificación. 71
3.4 LA CABECERA DE CARGA DE SEGURIDAD ENCAPSULADA (ESP). 72
3.4.1 Formato del Paquete de la Carga de Seguridad Encapsulada 73
3.4.2 Localización de la Cabecera ESP 76

4. REDES PRIVADAS VIRTUALES (VPN) 79

4.1 TÚNELES 80
4.1.1 Tipos de túneles: 81
4.2 COMPONENTES DE LA VPN 82
4.3 TIPOS DE VPN. 85
4.3.1 Tipos de IP VPN según RFC 2764. 85
4.3.2 Según el alcance de la VPN para la organización. 86
4.4 MODELOS DE IMPLEMENTACIÓN VPN 88
4.5 IMPLEMENTACIÓN DE LAS REDES PRIVADAS VIRTUALES 89
4.5.1 Implementación de VPN por Hardware 90
4.5.2 Implementación de VPN por Software 91
4.5.2.1 Tipos de VPN por Software 92
4.6 TRABAJANDO CON IPV6 95

5. INFRAESTRUCTURA DE IPSEC.....99

5.1 VENTAJAS Y LIMITACIONES 101
5.2 COMPONENTES DE IPSEC..... 102
5.2.1 Características de IPSec 104
5.3 CABECERAS Y ALGORITMOS. 105
5.4 MODOS TÚNEL Y TRANSPORTE..... 106
5.5 EL PROTOCOLO ISAKMP 107
5.6 INTERNET KEY EXCHANGE (IKE)..... 108
5.6.1 Fases de IKE 110
5.7 PROCESAMIENTO DE LOS PAQUETES DE ENTRADA Y SALIDA. 111
5.8 EL PROCESO DE NEGOCIACIÓN Y FILTRADO..... 113
5.9 DIRECTIVAS DE SEGURIDAD DE IP..... 113
5.9.1 ISAKMP y directivas de seguridad..... 114
5.9.2 Componentes de seguridad 114

6. FILTROS 117

6.1 MOTIVOS DE UN FILTRO 117
6.1.1 Reglas..... 118
6.2 ICMPv6 118
6.2.1 Formato de la cabecera ICMPv6..... 120
6.2.2 Mensajes ICMPv6..... 120
6.2.2.1 Mensajes de error..... 120
6.3 TIPOS DE FILTROS 123

7. FIREWALL 125

7.1 TIPOS DE FIREWALL..... 125
7.1.1 Firewall de capa de red 125
7.1.2 Firewall de capa de aplicación..... 126
7.2 VENTAJAS DE UN FIREWALL 126
7.3 FIREWALLS BASADOS EN LINUX 127
7.4 IPTABLES 130
7.5 NETFILTER6..... 131
7.6 IP6TABLES 131
7.6.1 Configuración de ipv6 en Red Hat Enterprise 132
7.6.2.1 Comandos de ip6table..... 134
7.6.2.2 Parámetros..... 136
7.6.2.3 Match Extensions..... 137
7.7.2.4 Instrucciones..... 140

8. CALIDAD DE SERVICIO (QOS) 143

8.1 BENEFICIOS DE QOS 143
8.2 ARQUITECTURAS DE QOS 144
8.3 CALIDAD DE SERVICIO EN IPV6 145
8.3.1. Campo Clase de Tráfico (8 bits) 145
8.3.2. Campo Etiqueta de Flujo (20 bits) 146

9. IMPLEMENTACIÓN DEL PROTOCOLO IPV6 CON DISPOSITIVOS CISCO 148

9.1 CONECTIVIDAD BÁSICA EN IPV6 152
9.2 VERIFICANDO LA CONECTIVIDAD BÁSICA EN IPV6. 154
9.3 CALIDAD DE SERVICIO PARA IPV6 156
9.3.1 Prerrequisitos para implementar QoS para Ipv6. 156
9.3.2 Estrategia para implementar calidad de servicio sobre Ipv6 157
9.3.3 Clasificación de paquetes en IPv6 158
9.3.4 Implementación calidad del servicio para IPv6 159
9.4 VPN CON IPSEC 163
9.4.1 Configurando IPsec de IPv6 en VTI (interfaz virtual del túnel) 166
9.4.2 Verificando el modo de configuración del Protocolo Ipsec en modo túnel 169

10. IMPLEMENTACIÓN DE IPV6 EN LOS SISTEMAS OPERATIVOS LINUX Y WINDOWS. 174

10.1 PROTOTIPO DE LA RED 174
10.1.1 Elementos a utilizar 176
10.2 GUÍA DE DIRECCIONAMIENTO IPV6 179
10.2.1 Direcciones para los hosts 181
10.2.2 Prefijos Generales IPv6 182
10.3 GUÍA DE INSTALACIÓN DEL PROTOCOLO IPV6 EN WINDOWS XP, WINDOWS 2000 Y LINUX 183
10.3.1 Instalación de IPv6 en plataforma Windows XP 183
10.3.1.1 Desde la ventana de comandos: 183
10.3.1.2 Desde el entorno de red: 186
10.3.2 Instalación de IPv6 en plataforma Windows 2000 188
10.3.3 Instalación de IPv6 en plataforma Linux Red HAT ES 4. 189
10.3.3.1 Comprobando el soporte del kernel para IPv6 189
10.3.3.2 Comprobando las herramientas de red para IPv6 189
10.3.3.3 Verificando el soporte de los scripts 191
10.4 GUÍA DE VERIFICACIÓN DE LA INSTALACIÓN EN WINDOWS XP, WINDOWS 2000 Y LINUX. 194
10.4.1 En plataforma Windows 2000 y XP 194
10.4.2 En plataforma Linux Red HAT ES 4. 195
10.5 GUÍA DE CONFIGURACIÓN BÁSICA DE IPV6 EN WINDOWS XP, WINDOWS 2000 Y LINUX. 195

10.5.1 En plataforma Windows XP 196
10.5.2 En plataforma Windows 2000 212
10.5.2 En plataforma Linux Red HAT ES 4. 214
10.5.2.1 Configuración de Linux con Zebra. 214
10.5.2.2 Pasos para configurar el router 219
10.5.2.3 Configuración de Linux, desde la consola. 225
10.5.2.4 Activando el servicio de ip6tables. 229
10.6 COMPROBACIÓN DE LA CONECTIVIDAD 233
10.6.1 En plataforma Windows XP 233
10.6.2 En Windows 2000. 247
10.6.3 En Linux Red HAT ES 4. 249
10.7 GUÍA DE CONFIGURACIÓN DE SERVICIOS HTTP, FTP EN LINUX 251
10.7.1 INSTALACIÓN Y CONFIGURACIÓN DE APACHE 2.2.4 251
10.7.2 INSTALACIÓN Y CONFIGURACIÓN DE FTP. 252
10.8 GUÍA DE IMPLEMENTACIÓN DEL FIREWALL EN LINUX. 254
10.9 GUÍA DE IMPLEMENTACIÓN DE CALIDAD DE SERVICIO (QOS) EN LINUX 265
10.9.1 FILTROS PARA QOS 267
10.10 GUÍA DE CONFIGURACIÓN DE UNA VPN. 272
10.10.1 Mecanismos de Transición 272
10.10.2 Túneles IPv6 sobre IPv4 273
10.10.2.1 Túneles 6in4 273
10.10.2.2 Túnel Broker 274
10.10.2.3 Túneles 6to4 275
10.10.2.4 Túneles Teredo 275
10.10.3 Establecimiento túnel 6in4 277
10.10.4 Configuración de 6in4 279

11. CONFIGURACIÓN DE DNS PARA IPV6 EN LINUX 283

11.1 COMPONENTES DE UN DNS. 284
11.2 ZONAS QUE SE PUEDEN RESOLVER 286
11.3 CARACTERÍSTICAS DE DNS PARA IPV6 287
11.4 EJEMPLO DE CONFIGURACIÓN 289

12. PROTOCOLOS DE ENRUTAMIENTO PARA IPV6. 292

12.1 PROTOCOLOS DE ENRUTAMIENTO EN DISPOSITIVOS CISCO 292
12.1.1 RIP Para Ipv6. 292
12.1.2 OSPF para IPv6 294
12.1.3 BGP para IPv6. 301
12.2 PROTOCOLOS DE ENRUTAMIENTO EN LINUX (DEMONIO QUAGGA) 304

12.2.1 Demonio de RIPngd 304
12.2.2 Demonio de OSPv3 305
12.2.3 Demonio de BGP-4 307

13. NAT EN IPV6 313

13.1 NAT EN IP6TABLES 313
13.2 NAT-PT 314
13.2.1 Cisco NAT-PT 315

RECOMENDACIONES 317

CONCLUSIONES 318

FUENTES DE INFORMACION 319

GLOSARIO 320

ANEXOS 327

ANEXO 1 MODELO OSI 328

ANEXO 2 CONFIGURACIÓN DE IPV6 EN WINDOWS XP 330

ANEXO 3 TUNNEL BROKERS 333

Introducción

El nuevo protocolo IP, se llama IPv6, el cual es toda una realidad, y actualmente muchas empresas, universidades, países ya lo están implementando. Existen muchos proyectos que pretenden dar a conocer el protocolo e implementarlo. IPv6 ofrece muchas características y beneficios adicionales que el actual ipv4, y una de las características más relevante es su forma y tamaño de la dirección IP.

En este documento se presenta la investigación del protocolo que inicia desde el formato de la cabecera ipv6 hasta temas avanzados como el uso ipv6 en un Firewall, Calidad de Servicio, VPN. Pero antes de llegar a esos temas avanzados, al inicio del documento se da a conocer la teoría, conceptos necesarios para comprender de una mejor manera el funcionamiento de ipv6.

Desde el capítulo dos al nueve se encuentra los temas teóricos como: introducción a IPv6, Seguridad en la red, mecanismos de seguridad, redes privadas virtuales, Filtros, Firewall y Calidad de servicio. Luego la implementación del protocolo se presenta a partir del capítulo diez, que trata sobre IPv6 con dispositivos CISCO.

Al final de este documento, en el capítulo once se presenta una serie de guías, las cuales presentan un escenario y distintas configuraciones y uso del protocolo ipv6, por ejemplo, configuración en Red Hat ES 4.0, Windows XP y Windows 2000, además se muestra también la configuración de ip6tables, QoS, VPN, utilizando IPv6 en Linux. Estos temas confirman la teoría proporcionada al inicio del documento.

En la sección de anexos, se presenta una explicación del MODELO OSI, este tema se menciona y se aborda en este trabajo, también puede encontrar una guía de comandos útiles para configurar Ipv6 en el sistema operativo Windows XP. Además se presenta una breve información sobre túneles brokers, el cual es una forma de obtener conectividad ipv6 cuando no se puede configurar ipv6 de forma nativo, o no se tienen las direcciones apropiadas para poder realizar pruebas con el protocolo.

Antecedentes

En nuestros días se ha incrementado aceleradamente la demanda de usuarios de Internet, y han surgido nuevas aplicaciones que requieren videoconferencia, multimedia, respuestas en tiempo real. El protocolo de Internet IPv4, presenta muchas limitaciones ante todo este desarrollo y crecimiento de la tecnología. Todos estos factores han provocado escasez de direcciones IP, tabla de ruteo ineficientes, incremento de redes que recurren a la herramienta NAT[1] (Network Address Translation) y un nivel bajo de seguridad.

Existe una nueva versión del protocolo IP (Internet Protocol), que está destinado a sustituir gradualmente al actual estándar IPv4. Este nuevo protocolo se llama IPv6. Existen muchas razones por las cuales se requiere de su investigación e implementación, como por ejemplo: incremento de direcciones, aplicaciones en tiempo real, Redes móviles, optimizar la calidad de servicio (QoS), inserción de nuevos dispositivos que utilizan IP, seguridad, etc.

La seguridad es uno de los puntos débiles de Internet, y ésta comúnmente se ve amenazada por muchos ataques de virus, fraudes, violación de la confidencialidad en la comunicación de datos de los usuarios, negación de servicio, etc. Con la utilización de IPv6, se puede utilizar mecanismos de seguridad y garantizar la comunicación de los datos.

Se puede pensar que el desarrollo y ejecución de IPv6 está muy lejos, pero es importante destacar que existe una brecha digital que se está desplegando rápidamente, es por eso que muchos países se han dedicado a investigar sobre IPv6, han unido esfuerzos, para dar aportes y ayudar a la transición de IPv4 a IPv6. Gracias al interés y la necesidad de la implementación del nuevo protocolo, se han logrado muchos avances nacionales e internacionales, entre ellos se puede mencionar:

[1] *Es un estándar que utiliza una o más direcciones IP para conectar varias computadoras a otra red, los cuales tienen una dirección IP completamente distinta*

1. El gobierno federal de Estados Unidos ha declarado que para el 2008 todas sus agencias trabajarán sólo con IPv6.
2. El 6 de Junio del 2006 se estableció como la fecha en que Internet inicia la transición oficial a IPv6, Después de más de diez años de planificación, desarrollo y experiencia con IPv6, por disposición de IETF2 concluye el proyecto de 6Bone, después de haber tenido como propósito estimular el despliegue y experiencia de IPv6.
3. En marzo del 2006 se instalan los nodos de Internet2 en las universidades UCA, UES, UFG, ITCA, UNICO, UTEC y UDB, los cuales trabajan nativamente con IPv6.

[2] *IETF (Grupo de Trabajo en Ingeniería de Internet).*

Importancia de la investigación.

El tiempo ha transcurrido desde que se diseñó e implementó IPv4 en las redes locales, e Internet, actualmente existe una gran demanda de usuarios de Internet, uso de PDA3, teléfonos celulares, redes móviles, aparatos industriales que requieren de direcciones de IP, empresas de manufactura, electrodomésticos, ante esta situación muchos han pronosticado que dentro de unos pocos años sufriremos de una escasez de direcciones IP y podría causar muchos problemas en general.

Debido a esta problemática, muchos investigadores y desarrolladores, aproximadamente desde hace unos diez años han estado trabajando en una nueva versión del protocolo de Internet (IP). Si hablamos de nuevas tecnologías a implementar, podemos abordarlas de manera general, pero existe un área primordial para el funcionamiento de una red, esto es la seguridad. A continuación se mencionan unos aspectos sobre la importancia de la realización de este proyecto:

- La temática se trata sobre IPv6, y el tema de la seguridad con este proyecto, el cual es un nuevo protocolo, por lo tanto en nuestro país, no se cuenta con manuales accesibles para el público en general que esté interesado en documentarse sobre el nuevo protocolo. Si existe información pero es solamente a nivel interno de empresas interesadas en implementar el protocolo.

- Se puede pensar que falta mucho tiempo para que se IPv6 se comience a implementar en nuestro país, pero la realidad es otra, en muchos países latinoamericanos durante todos estos años se han estado preparando para la transición de IPv4 a IPv6.

- El principal motivo para cambiar de IPv4 a IPv6 es la cantidad de direcciones, ya que se plantea la posibilidad de asignar direcciones IP a dispositivos tales como: celulares, cámaras fotográficas, electrodomésticos, agendas, computadoras de vehículos, etc. Incrementando de manera exponencial el uso de las direcciones, lo que es imposible con IPv4.

[3] ***P**ersonal **D**igital **A**ssistant, (Ayudante personal digital)*

- Pronto las empresas comerciales, estatales, educativas, etc., se verán en la necesidad de realizar ésta transición también.

- Otra razón importante para llevar a cabo el proyecto, es que el documento servirá como una referencia o material de apoyo, para todos aquellos que estén interesados en revisar y actualizar los conocimientos de seguridad, pero ahora enfocados en IPv6. La seguridad es un aspecto primordial que se debe de implementar en todo diseño de una red, y el proyecto que se propone demostrará las principales funciones, y aplicaciones como lo son Redes privadas virtuales (VPN), Firewalls, y Calidad de Servicio (QoS).

- La seguridad en IPv4 es opcional, existen herramientas de seguridad, que se pueden agregar al protocolo, como lo es IPSec. Originalmente cuando se creó IPV4 se diseñó para redes aisladas. Mientras que en IPv6 el protocolo IPSec se encuentra integrado dentro de la cabecera.

El protocolo IPv6 integra en su cabecera dos nuevos campos (Clase de Tráfico y etiqueta de flujo), estos campos permiten una de las características fundamentales de IPv6: Calidad de Servicio (QoS), y un mecanismo de control de flujo de datos.

Situación actual de IPv6 en El Salvador.

En nuestro país, existe una organización llamada RAICES, que es la Red Nacional de Investigación y Educación de El Salvador (NREN[4]), es miembro fundador de CLARA (Cooperación Latinoamericana de Redes Avanzadas) y socio local de DANTE (Delivering Advanced Network To Europe) para el Proyecto ALICE (América Latina Interconecta con Europa).

Redes Avanzadas:

- Son redes paralelas a las comerciales para dar conectividad exclusiva a las instituciones de educación e investigación.

- Son una infraestructura global de información y comunicación de gran capacidad.

- Son plataformas de pruebas experimentales de nuevos servicios y tecnologías avanzadas.

- Facilitan la investigación y refuerzan la colaboración entre equipos ubicados en distintos puntos del planeta.

- Permiten formar conglomerados de instituciones que forman verdaderos "institutos de investigación virtuales".

A continuación se explican en que consisten las organizaciones DANTE, ALICIA y CLARA:

DANTE (entrega de la tecnología de red avanzada a Europa) planea, construye y opera con las redes avanzadas para la investigación y la educación. DANTE proporciona la infraestructura de las comunicaciones de datos esencial al desarrollo de la comunidad de investigación global.

[4] *NREN (national research and education networks, recursos nacionales y educación de redes).*

ALICIA (América Latina Interconectada Con Europa), éste proyecto inicio en el año 2003 para desarrollar la red de Red CLARA, que proporciona la infraestructura de la red de la investigación del IP dentro de la región latinoamericana y hacia Europa. Es manejado por DANTE, y es el 80% financiado por la Comisión de las Comunidades Europeas.

La organización CLARA (Cooperación Latinoaméricana en Redes Avanzadas), tiene los siguientes objetivos:

- Coordinación entre las NRENs de Latinoamérica y otros actores
- Cooperación para promover el desarrollo científico y tecnológico
- Planeamiento e Implementación de una red regional de características avanzadas para interconectar las NRENs de Latinoamérica
- Interconexión de la región con las redes avanzadas del resto del mundo

Actualmente en El Salvador, el protocolo IPv6, se encuentra en etapa de investigación por parte de la organización RAICES, es importante mencionar, que otros miembros de CLARA están brindando muchos aportes investigativos para el desarrollo de IPv6. Como por ejemplo:

- RAAP (Red académica Peruana) http://www.raap.org.pe/
- RNP (Red Nacional de enseñanza e Investigación, Brasil) http://www.rnp.br/
- CUDI (Corporación Universitaria para el desarrollo de Internet, México) www.cudi.edu.mx
- RAU (Red académica Uruguay, Uruguay) http://www.rau.edu.uy/redavanzada/

Los miembros de RAICES son los siguientes:

- Universidad Centroamericana José Simeón Cañas (UCA)
- Universidad de El Salvador (UES)
- Universidad Don Bosco (UDB)
- Universidad Francisco Gavidia (UFG)
- Universidad Tecnológica (UTEC)
- Universidad Católica de Occidente (UNICO)
- Instituto Tecnológico Centroamericano (ITCA)

Capítulo

1

Introducción a IPv6

Temas:

- Introducción a IPv6.
- Limitaciones de IPv4.
- Características de IPv6.
- Diferencias entre IPv4 e IPv6.
- Terminología de IPv6.
- Beneficios e innovación.
- Cabecera de IPV6.
- Direccionamiento en IPv6.
- Configurando IPv6 en los Sistemas Operativos.

1. Introducción a IPv6

¿Qué es IPv6?

IPv6 es la versión 6 del Protocolo de Internet, es un estándar del nivel de red encargado de dirigir y encaminar los paquetes a través de una red. Está destinado a sustituir al estándar IPv4, cuyo límite en el número de direcciones de red admisibles está empezando a restringir el crecimiento de Internet y su uso. Adoptado por el IETF en 1994 (cuando era llamado "IP Next Generation" o IPng), IPv6 cuenta con un pequeño porcentaje de las direcciones públicas de Internet, que todavía están dominadas por IPv4. La adopción de IPv6 ha sido frenada por la implementación de NAT, que alivia parcialmente el problema de la falta de direcciones IP. Pero NAT hace difícil y en algunos casos imposible el uso de algunas aplicaciones P2P (Peer to peer), como son la voz sobre IP (VoIP).

1.1 Limitaciones de IPv4

La versión actual del Protocolo de Internet IP, conocida como versión cuatro o IPv4 (RFC[5] 791), no ha cambiado mucho desde su creación en 1981. El crecimiento del protocolo ha sido detenido ante el incremento del uso de la red a nivel mundial y presenta las siguientes limitantes:

✓ **Agotamiento de las direcciones IPv4 ante el actual crecimiento de Internet:**
Aunque los 32 bits de espacio para las direcciones de IPv4 permiten 4, 294, 967,296 direcciones, se prevee que existirá un agotamiento de las direcciones para los próximos años. Ante este problema de escasez de direcciones, se creó NAT. Aunque permiten más clientes conectados a Internet, el proceso de conexión genera problemas de cuello de botella y se bloquean algunos tipos de comunicaciones.

[5] *Request For Comment, es un conjunto de notas técnicas y organizativas donde se describen los estándares o de Internet*

✓ **La necesidad de una configuración simple:**

La mayoría de las implementaciones de IPV4 son configuradas manualmente o se utiliza un protocolo de configuración de direcciones como por ejemplo DHCP[6], si se tienen muchos hosts y dispositivos que requieren de IP, entonces se necesita de una configuración automática de direcciones y otras herramientas que no se basen en la infraestructura de DHCP.

✓ **Requerimientos de seguridad en el nivel de IP:**

El establecimiento de la comunicación privada sobre un medio público, como lo es Internet requiere de servicios de criptografía, para brindar protección en el transporte de los datos desde el origen hacia el destino. En IPv4 se puede implementar el Protocolo de Seguridad o IPSec; éste estándar es opcional, es decir que es un agregado, originalmente no estaba contemplado en el diseño del protocolo IPv4.

✓ **Necesidad de brindar soporte para aplicaciones de datos en tiempo real, llamado también calidad de servicio (QoS):**

Aunque existe un estándar de QoS para IPv4, el soporte para las aplicaciones en tiempo real se basa en 8 bits del Tipo de Servicio (TOS), que es un campo de identificación de ejecución. Desafortunadamente la funcionalidad es limitada. IPv6 ha sido diseñado con el prepósito de minimizar el trabajo de las capas superiores y proporcionar nuevas funcionalidades.

1.2 Características de IPv6

Existen cambios en la funcionalidad de IPv6, algunos de ellas son los siguientes:

- **Tamaño de la dirección:** la longitud del campo de la dirección aumenta, de 32 bits a 128 bits.

- **Formato de la cabecera:** es completamente diferente a la de IPv4, algunos de los campos de la cabecera de IPv4 como CheckSum, IHL, identificación de bandera, no aparecen en la cabecera de IPv6.

[6] *Protocolo de red que permite a los nodos de una red IP obtener sus parámetros de configuración automáticamente.*

- **Extensión de la cabecera**: IPv6 codifica la información en cabeceras diferentes.

- **Etiqueta de flujo**: IPv6 agrega una "etiqueta de flujo" (flow label) con el objetivo de identificar el flujo de ciertos paquetes que requieren de calidad de servicio.

- **Proceso de fragmentación**: este proceso solo se realiza desde el origen de la transmisión; por lo tanto, los enrutadores no ejecutan paquetes de fragmentación, si un paquete necesita ser fragmentado, el origen revisa la cantidad de MTU[7] y decide si se debe de fragmentar el paquete.

- **Soporte para audio y video**: IPv6 incluye un mecanismo, que permite establecer entre el emisor y receptor una ruta de alta calidad. Dicha ruta se utiliza para el uso de audio conferencia y aplicaciones de videos que requieran de garantía durante el desempeño.

- **No se establece el campo CheckSum**: este campo no se incluye en la cabecera de IPv6, la finalidad de no tomarlo en cuenta es reducir el tiempo del proceso del paquete en un enrutador.

- **Protocolo extensible**: en el protocolo de IPv6 no se especifican todas las características del protocolo, el esquema de extensión hace flexible IPv6 de IPv4, si se generan nuevas características pueden ser agregadas al diseño.

- **Seguridad**: IPv6 tiene incorporada las características de autenticación y la confidencialidad dentro de la cabecera.

[7] *Unidad Máxima de Transferencia*

1.3 Diferencias entre IPv4 e IPv6

Existen algunas diferencias importantes en el direccionamiento de IPv6 respecto de IPv4, a continuación se mencionan algunas de ellas:

No hay direcciones broadcast, su función es sustituida por direcciones multicast. Las direcciones IPv6, indistintamente de su tipo (unicast, anycast o multicast), son asignadas a interfaces, no a nodos. Esto se debe a que una interfaz pertenece a un único nodo. Todas las interfaces deben de tener, al menos, una dirección unicast. La siguiente tabla muestra en forma comparativa las diferencias más significativas de IPv4 e IPv6.

Diferencias	IPv4	IPv6
Direcciones	Las direcciones de origen y destino tienen una longitud de 32 bits (4 bytes).	Las direcciones de origen y destino tienen una longitud de 128 bits (16 bytes).
IPSec	La compatibilidad es opcional.	La compatibilidad es obligatoria
Identificación del número de paquetes	No existe ninguna identificación de flujo de paquetes para que los enrutadores controlen la calidad de servicio (QoS) en el encabezado IPv4.	Se incluye la identificación del flujo de paquetes para que los enrutadores controlen la QoS en el encabezado IPv6, utilizando el campo Flow Label (etiqueta de flujo).
Fragmentación	La llevan a cabo los enrutadores y el host que realiza el envío.	No la llevan a cabo los enrutadores, sino, únicamente el host que realiza el envío.
Encabezado	Incluye una suma de comprobación.	No incluye una suma de comprobación.
Opciones	El encabezado lo incluye.	Todos se trasladan a los encabezados de extensión IPv6.
Marcos de solicitud ARP	El Protocolo de resolución de direcciones (ARP) utiliza los marcos de solicitud ARP de	Los marcos de solicitud ARP se sustituyen por mensajes de solicitud de vecinos de

	difusión para resolver una dirección IPv4 como una dirección de capa de vínculo.	multidifusión.
Administrar la pertenencia a grupos locales de subred	Se utiliza el Protocolo de administración de grupos de Internet (IGMP).	IGMP se sustituye con los mensajes de Descubrimiento de escucha de Multidifusión.
Determinar la dirección IPv4 de la mejor puerta de enlace predeterminada	Se utiliza el Descubrimiento de enrutadores ICMP, y es opcional.	El Descubrimiento de enrutadores ICMP queda sustituido por la Solicitud de enrutadores ICMPv6 y los mensajes de anuncio de enrutador, y es obligatorio.
Direcciones de multidifusión	Se utilizan para enviar tráfico a todos los nodos de una subred.	No hay direcciones de multidifusión IPv6. De forma alternativa, se utiliza una dirección de multidifusión para todos los nodos de ámbito local del vínculo.
DNS	Utiliza registros de recurso (A) de dirección de host en el Sistema de nombres de dominio (DNS) para correlacionar nombres de host con direcciones IPv4.	Utiliza registros de recurso (AAA) de dirección de host en el Sistema de nombres de dominio (DNS) para correlacionar nombres de host con direcciones IPv6.
Direcciones IP relacionados con host	Utiliza registros de recurso (A) de puntero en el dominio DNS IN-ADDR. ARPA para correlacionar direcciones IPv4 con nombres de host.	Utiliza registros de recurso (PTR) de puntero en el dominio DNS IP6.INT para correlacionar direcciones IPv6 con nombres de host.
Tamaño de paquete	Debe admitir un tamaño de 576 bytes (posiblemente fragmentado).	Debe admitir un tamaño de 1280 bytes (sin fragmentación).

Tabla 1 Comparación entre IPv4 e IPv6

1.4 Terminología de IPv6

En los siguientes capítulos se utilizan términos fundamentales, para poder conocer la terminología adecuada, a continuación se proporciona una lista de conceptos, En la Figura 1 se presenta una red IPv6.

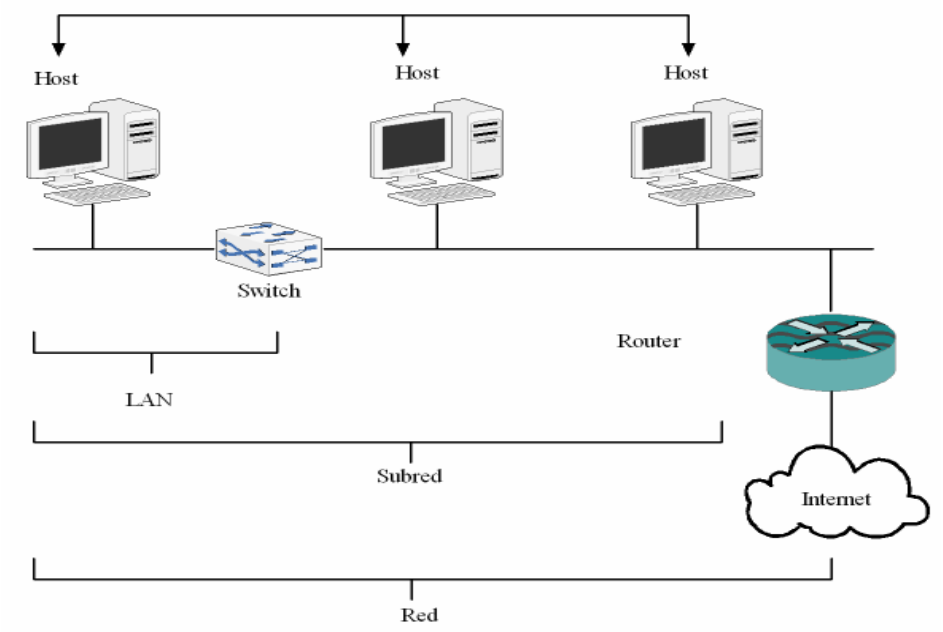


Figura 1 Elementos de una red de IPv6

Nodo: cualquier dispositivo que ejecuta una implementación de IPv6, Este término incluye a enrutadores y hosts.

Enrutador: es un nodo que puede encaminar los paquetes IPv6 no necesariamente posee una dirección IP. En una red basada en IPv6, un enrutador también administra la información y configuración de los host que pertenecen a la red.

Host: es un nodo que no puede enviar paquetes IPv6, no es necesario que posea una dirección IP. En el flujo del tráfico de datos, un host puede ser origen (emisor) o destino (receptor). Los Hosts descartan el tráfico que reciben y que no les pertenece.

Red de área local (LAN): Una red local es la interconexión de varios ordenadores y periféricos. Su extensión está limitada físicamente a un edificio o a un entorno de unos pocos kilómetros. Su aplicación más extendida es la interconexión de ordenadores personales y estaciones de trabajo en oficinas, fábricas, etc. Entre las tecnologías de enlace a nivel de LAN, se mencionan: Ethernet, Token Ring, FDDI (Fiber Distributed Data Interface). Si la red es de tipo WAN, entonces las tecnologías que se utilizan son: el protocolo PPP (Point to Point), Frame Relay, y ATM (Asynchronous Transfer Mode).

Enlace: se le llama enlace a la conexión de uno o más segmentos que pertenecen a una red, dicha conexión se realiza a través de los enrutadores.

Subnet (SubRed): es uno o más enlaces que utilizan la misma cantidad de bits en la parte del prefijo de la dirección, una dirección esta compuesta de dos partes: red y host.

Red: dos o más subredes conectadas por medio de enrutadores.

Vecinos (Neighbors): son los enrutadores que se encuentran conectados en el mismo enlace.

Dirección: es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo dentro de una red que utilice el protocolo IP; dicho número no se ha de confundir con la dirección MAC que es un número hexadecimal fijo que es asignado a la tarjeta o dispositivo de red por el fabricante, mientras que la dirección IP se puede cambiar.

MTU: Simboliza la cantidad máxima de paquetes representados por octetos (bytes) y posteriormente pueden ser desfragmentados sobre el enlace.

1.5 Beneficios e innovación

Debido a las diferencias significativas entre los protocolos IPv4 e IPv6. El nuevo protocolo presenta los siguientes beneficios:

- **Configuración automática:** IPv6 aplica a la red el principio *"plug and play*". Un sistema recién instalado puede integrarse sin problemas en la red (local). El mecanismo automático de configuración de la Terminal deduce la propia dirección de la información transmitida a través del protocolo NDP ("Neighbor Discovery Protocol") por los enrutadores adyacentes. Este procedimiento no requiere la intervención del administrador y tiene la ventaja adicional de hacer el mantenimiento de un servidor central con las direcciones disponibles.

- **Movilidad**: IPv6 permite asignar varias direcciones paralelas a una interfaz de red. Se puede comparar este mecanismo con el "roaming" de las redes de telefonía móvil.

- **Comunicación segura:** mientras que en IPv4 la comunicación segura constituía una función adicional, IPv6 incluye IPSec y por tanto la comunicación segura entre dos sistemas mediante un túnel a través de Internet.

- **Compatibilidad con la versión anterior**: no es realista creer que la migración de la totalidad de Internet de IPv4 a IPv6 se va a llevar a cabo rápidamente. Por eso es importante que ambas versiones puedan coexistir en Internet e incluso en un mismo sistema. La coexistencia de ambos protocolos en Internet está asegurada por el uso de direcciones compatibles (las direcciones IPv4 pueden convertirse fácilmente a direcciones IPv6).

- **Multicasting:** mientras que en IPv4 algunos servicios envían por broadcast sus paquetes a todos los miembros de la red local; IPv6 permite un procedimiento muy distinto: con multicast es posible dirigirse al mismo tiempo a un grupo de ordenadores. Es decir, no a todos (broadcast) o sólo a uno (unicast), sino a un grupo.

1.6 La cabecera de IPV6

El Protocolo IP pertenece al nivel de red, por lo tanto, es utilizado por los protocolos del nivel de transporte como TCP para encaminar los datos hacia su destino. IP tiene únicamente la misión de encaminar el datagrama. Para ello se utiliza una cabecera que se antepone al datagrama que se está tratando y contiene información para IP, y unos datos que son relevantes sólo para los protocolos de más alto nivel.

1.6.1 Estructura del paquete de IPv6.

Un paquete de IPv6 consiste de una cabecera IPv6, extensión de la cabecera y Unidad de datos de protocolo de la capa superior. En la figura 2 se presenta la estructura de un paquete de IPv6.

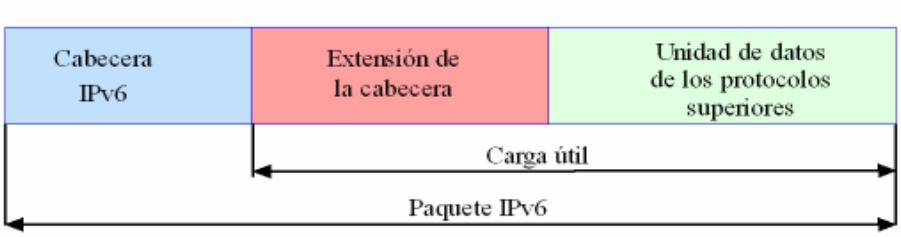


Figura. 2 Estructura de un paquete IPV6.

Los componentes de un paquete de IPv6 son los siguientes:

Cabecera IPv6: la cabecera de IPv6 siempre se encuentra presente dentro del paquete, y esta compuesta de 40 bytes.

Extensión de la cabecera: la cabecera de IPv6 y las extensiones de su cabecera, reemplazan a la cabecera existente de IPv4 y sus opciones. El formato de las nuevas extensiones de la cabecera permite a IPv6 soportar nuevas necesidades y capacidades. A diferencia de IPv4 no tienen un tamaño máximo y permiten la expansión de la extensión de los datos que se necesiten durante la comunicación.

Unidad de datos de protocolo de la capa superior: consisten en un conjunto de protocolos que pertenecen a la capa superior, como por ejemplo: ICMPv6, segmentos TCP o mensajes UDP. Cuando los paquetes IPv6 son ejecutados, se combinan las extensiones de la cabecera con las unidades de protocolo de la capa superior.

1.6.2 Estructura general de la cabecera

Existen cinco campos dentro de la cabecera de IPv4, que ya no pertenecen a la cabecera de IPv6, estos son*: longitud de la cabecera, identificación, bandera, Fragmentación Offset, Checksum*.

El campo de **la longitud de la cabecera**, no ha sido tomado en cuenta, por que la cabecera de IPv6 es de longitud completa. En IPv4 el tamaño mínimo de la longitud es de 20 bytes, si hay opciones adicionales, entonces son agregadas, y el tamaño se incrementa a 60 bytes. Por consiguiente con IPv4, el tamaño de la longitud de la cabecera es muy importante, en IPv6 las opciones son definidas por las extensiones de la cabecera.

Los campos: **identificación, banderas, y fragmentación Offset** son manejados en la fragmentación de un paquete con la cabecera de IPv4. El proceso de fragmentación ocurre si un paquete tiene un tamaño largo y es enviado sobre una red que solo soporta tamaño de paquetes pequeños. En este caso, los enrutadores que trabajan con IPv4 dividen el paquete en pequeños pedazos y luego los envían como múltiples paquetes. El host destino colecciona todos los paquetes y después los reensambla. Si un host origen quiere fragmentar un paquete, tendrá que utilizar la extensión de la cabecera para poder realizarlo.

El campo **ChecKSum** ha sido removido para proveer un proceso rápido. El proceso de checksum, es realizado desde el nivel de acceso al medio, IP es un protocolo de mejor esfuerzo, entonces la integridad de los datos es responsabilidad de las capas superiores. En la figura 3, se muestran los campos que mantienen el nombre de IPv4 en IPv6, así como los que han sido eliminados, los que han cambiado de nombre y los campos nuevos dentro de IPv6.

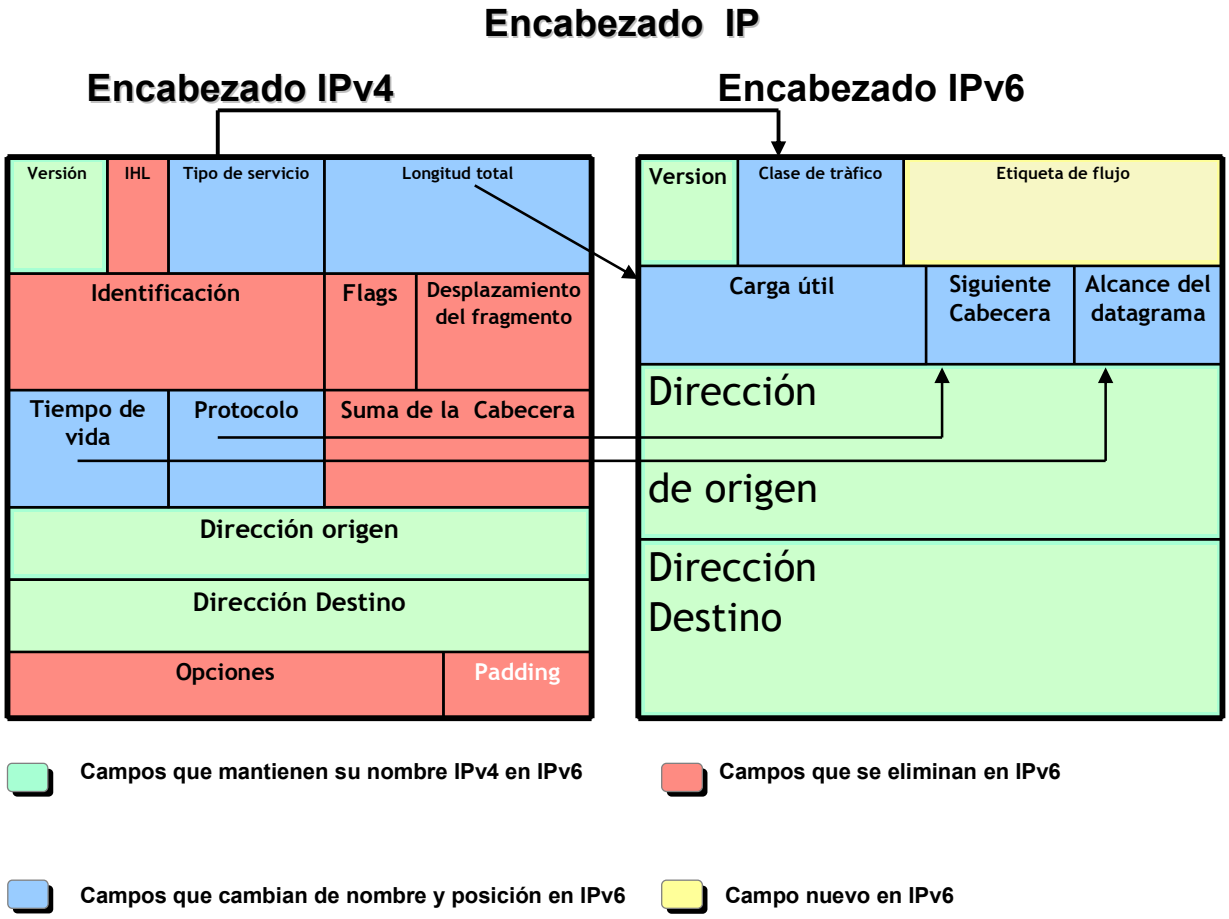


Figura 3 Comparación entre las cabeceras de IPv4 e IPv6

1.6.3 Los campos en la cabecera de IPv6

Para comenzar a familiarizarse con los campos de la cabecera IPv6, primero se debe de comprender el funcionamiento de IPv6, La cabecera de IPv6 ha eliminado los campos que no son realmente necesarios y se han agregado campos que proveen mejor soporte para el trafico de tiempo real. En la figura 4, se muestra la estructura de la cabecera IPv6, en base a la información que se encuentra en la RFC 2460[8].

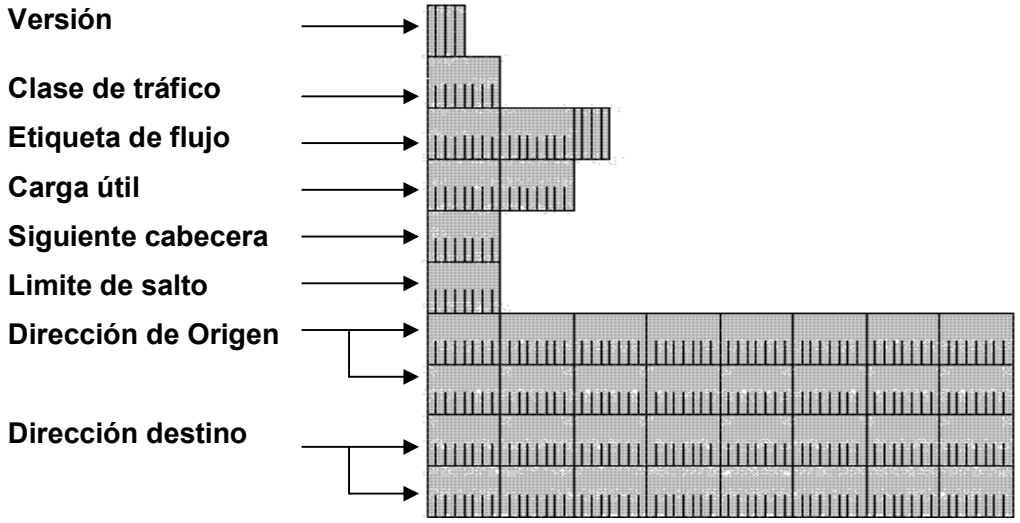


Figura 4 Estructura de la cabecera IPv6

Los campos son los siguientes:

Versión (4 Bits)

Indica la versión del protocolo IP. El tamaño de este campo es de 4 bits. Se evalúa si no existe transición de un paquete que pertenezca a IPv4 hacia IPv6.

[8] *RFC 2460, "Especificaciones del Protocolo Internet Versión 6 (IPv6)"*

Clase de tráfico (1 Byte)

Identifica la clase del paquete IPv6 o la prioridad que posee. El tamaño es de 8 bits. Este campo es una de las nuevas aportaciones para establecer algunos tipos de aplicaciones que se puedan ejecutar en tiempo real, como por ejemplo: videoconferencia, telefonía. Este campo provee funcionalidad igual al campo de tipo de servicio de IPv4.

Etiqueta de flujo (20 Bits)

Por medio de la etiqueta del flujo se detecta la secuencia de un paquete que se ejecuta entre la comunicación de un nodo origen y un nodo destino. El tamaño del campo es de 20 bits. La etiqueta de flujo es usado por las conexiones que no utilizan por defecto la calidad de servicio (QoS), tales como la necesidad de aplicaciones de tiempo real (voz y video).

Carga útil (2 Bytes)

El tamaño de este campo es de 16 bits. Este campo incluye la extensión de las cabeceras de las capas superiores. Se pueden almacenar más de 65,535 bytes. Si se detecta una cantidad mayor a 65,535.

Siguiente cabecera (1 Byte)

Indica el tipo de extensión de la cabecera, de las capas superiores (tales como TCP, UDP, o ICMPv6). El tamaño del campo es de 8 bits. En IPv4, este campo es el campo *tipo de protocolo*. Si la próxima cabecera es UDP o TCP, el campo contendrá el número igual al del protocolo, así como en IPv4, por ejemplo el protocolo numero 6 para TCP o 17 para UDP. Pero si las extensiones de la cabecera son usadas con IPv6, entonces contendrá el tipo de la próxima extensión de la cabecera. En la siguiente tabla[9] , se muestran los números de los protocolos asignados, con sus respectivas referencias.

[9] *La tabla 3 muestra algunos de los protocolos, para mayor información y obtener el listado completo de los protocolos, puede citar la siguiente dirección http://www.iana.org/assignments/protocolnumbers.*

Valor (Decimal)	Protocolo		Referencia
0	HOPOPT	IPv6 Hop-by-Hop Option	RFC1883
1	ICMP	Internet Control Message	RFC792
2	IGMP	Internet Group Management	RFC1112
3	GGP	Gateway-to-Gateway	RFC823
4	IP	IP in IP (encapsulation)	RFC2003
6	TCP	Transmission Control	RFC793
8	EGP	Exterior Gateway Protocol	RFC888
9	IGP	any private interior gateway	IANA
17	UDP	User Datagram	RFC768,JBP
41	IPv6	IPv6	Deering
43	IPv6-Route	Routing Header for IPv6	Deering
44	IPv6-Frag	Fragment Header for IPv6	Deering
45	IDRP	Inter-Domain Routing Protocol	Sue Hares
46	RSVP	Reservation Protocol	Bob Braden
50	ESP	Encrypted Security Payload	RFC2406
51	AH	Authentication Header	RFC2402
58	IPv6-ICMP	ICMP for IPv6	RFC1883
59	IPv6-NoNxt	No Next Header for IPv6	RFC1883
60	IPv6-Opts	Destination Options for IPv6	RFC1883
88	EIGRP	EIGRP	CISCO,GXS
108	IPComp	IP Payload Compression Protocol	RFC2393
115	L2TP	Layer Two Tunneling Protocol	Aboba
132	SCTP	Stream Control Transmission Protocol	Stewart
134-254	No asignado		RFC3692
255	Reservado		IANA

Tabla 2 Valores de los protocolos que pertenecen a la próxima cabecera.

Limite de salto (1 Byte)

Indica el número máximo de saltos que se puede realizar. Este campo es el equivalente al tiempo de vida (TTL) de la versión 4 de IP.

Dirección de Origen (16 Bytes)

El campo de la dirección de origen indica la dirección IPv6 del host origen, el tamaño de este campo es de 128 bits.

Dirección destino (16 Bytes)

El campo de la dirección destino indica la dirección IPV6 del nodo que es destino, El tamaño del campo es de 128.

1.6.4 Cabeceras de extensión de IPv6

El uso de un formato flexible de cabeceras de extensión es una idea innovadora que permite ir añadiendo funcionalidades de forma paulatina. Este diseño aporta gran eficacia y flexibilidad, se pueden definir en cualquier momento a medida que se vayan necesitando entre la cabecera fija y la carga útil. La RFC 2460[10] específica las siguientes extensiones que deben ser soportadas por todos los nodos IPv6, se recomienda que la extensión de la cabecera sea colocada después de la cabecera principal de IPv6 según el siguiente orden:

1. Encabezado IPv6.
2. Cabecera de opciones de salto a salto.
3. Cabecera de encaminamiento.
4. Cabecera de fragmentación.
5. Cabecera de autenticación.
6. Cabecera de encapsulado de seguridad de la carga útil.
7. Cabecera de opciones para el destino

[10] *RFC 2460, "Especificaciones del Protocolo Internet Versión 6 (IPv6)"*

- **Encabezado principal:** Cabecera principal de tamaño fijo de 40 octetos.

- **Cabecera de opciones de salto a salto (Hop-by-Hop):** transporta información opcional, contiene los datos que deben ser examinado por cada nodo a través de la ruta de envió de un paquete. Su código es 0.

- **Cabecera de encaminamiento (Routing):** se utiliza para que un origen IPv6 indique uno o más nodos intermedios que se han de visitar en el camino del paquete hacia el destino. El código que utiliza es 43.

- **Cabecera de fragmentación (Fragment):** hace posible que el origen envíe un paquete más grande de lo que capacidad de MTU de la ruta. Se debe de tener en cuenta que al contrario que en IPv4, en IPv6 la fragmentación de un paquete solo se puede realizar en los nodos de origen. El código empleado en esta cabecera es 44.

- **Cabecera de autenticación (Authentication Header):** se utiliza para proveer servicios de integridad de datos, autenticación del origen de los datos. El código de ésta cabecera es 51.

- **Cabecera de encapsulado de seguridad de la carga útil (Encapsulating Security Payload):** permite proveer servicios de integridad de datos. El código al que hace referencia esta cabecera es el 50.

- **Cabecera de opciones para el destino (Destination):** se usa para llevar información opcional que necesita ser examinada solamente por los nodos destino del paquete. La última de las cabeceras utiliza el código 60.

Cada extensión posee 64 bits (8 byte). En la figura 5 se muestra los enlaces de los puntos formados por la próxima cabecera de varios paquetes IPv6.

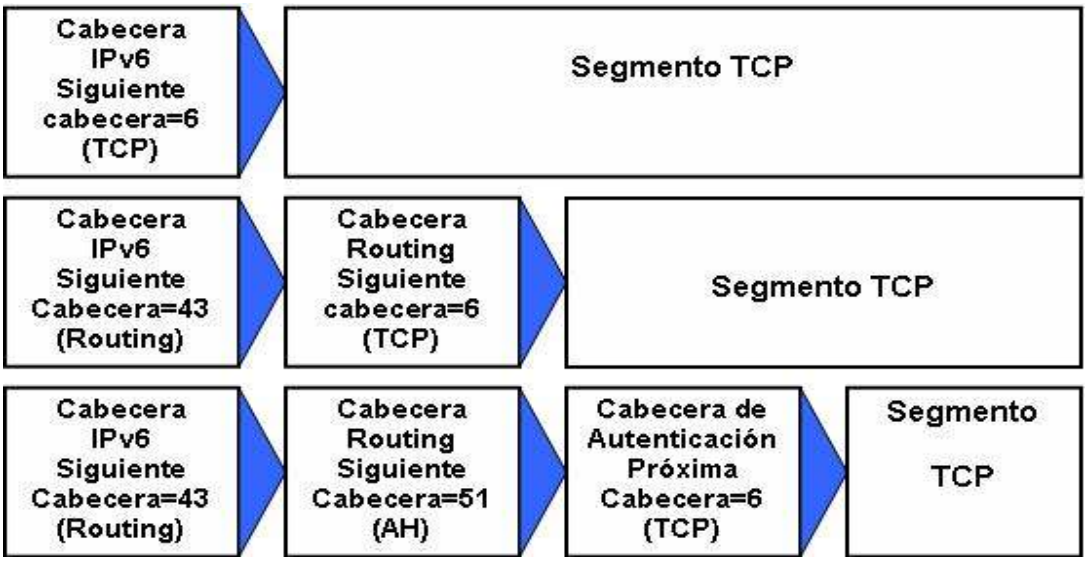


Figura 5. Unión de puntos formados por el campo de la próxima cabecera.

1.7 Direccionamiento en IPv6

Las direcciones IPv6 poseen son identificadores de 128 bits [11] de longitud. Identifican interfaces de red. A una interfaz de un nodo se le pueden asignar múltiples direcciones; dichas direcciones se clasifican en tres tipos:

- **Unicast:** identificador para una única interfaz. Un paquete enviado a una dirección unicast es entregado solo a la interfaz identificada con dicha dirección. Es equivalente a las direcciones IPv4 actuales.

- **Anycast**: Identificador para un conjunto de interfaces (típicamente pertenecen a diferentes nodos). Un paquete enviado a una dirección anycast es entregado en cualquiera de las interfaces identificadas con dicha dirección (la que se encuentre más cerca). Nos permite crear, por ejemplo, ámbitos de redundancia, de forma que varias máquinas puedan ocuparse del mismo tráfico según una secuencia determinada.

[11] *RFC 2373 Arquitectura del direccionamiento IP versión 6*

- **Multicast:** Identificador para un conjunto de interfaces (por lo general pertenecientes a diferentes nodos). Un paquete enviado a una dirección multicast es entregado a todas las interfaces identificadas por dicha dirección.

1.7.1 Direcciones Unicast

Las direcciones unicast, son agregables con máscaras de bits contiguos, similares al caso de IPv4, con CIDR (Class-less interdomain Routing). Los nodos IPv6 pueden no tener ningún conocimiento o mínimo de la estructura interna de las direcciones IPv6, dependiendo de su misión en la red. Pero como mínimo, un nodo debe de considerar que las direcciones unicast (incluyendo la propia), no tienen estructura:

Figura 6 Estructura de dirección IPv6.

Un host más sofisticado, conocería el prefijo de la subred, del enlace al que está conectado:

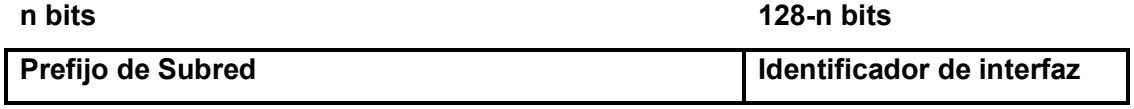


Figura 7 Prefijos de subred

El "identificador de interfaz" se emplea, por tanto, para identificador interfaces en un enlace, y deben ser únicos en dicho enlace. En muchos casos también serán únicos en un ámbito más amplio. Por lo general, el identificador de interfaz coincidirá con la dirección de la capa de enlace de dicha interfaz (MAC Address[12]). El mismo identificador de la interfaz puede ser empleado en múltiples interfaces del mismo nodo, sin afectar a su exclusividad global en el ámbito IPv6. Existen cinco tipos de direcciones Unicast, las cuales se detallan a continuación:

[12] *Dirección MAC (Media Access Control Address)*

1. Dirección no especificada (Unespecified Address): Está compuesta por 16 bytes nulos **(0:0:0:0:0:0:0:0)** y sólo puede utilizarse como dirección inicial; mientras se inicializa y se recibe una dirección fija. También puede utilizarse para funciones internas que requieran la especificación de una dirección IP.

2. Dirección interna (Loopback Address): Se define como 15 bytes nulos y un byte con el último bit a 1 (0:0:0:0:0:0:0:1). Esta dirección es interna y de ninguna forma puede circular por la red o ser dirección de origen o destino de un datagrama. Su utilidad viene dada para los ordenadores que no dispongan de una conexión de red y deseen simular el comportamiento de conexión a una red mediante una dirección fantasma que nunca saldrá del propio ordenador.

3. Direcciones tipo IP versión 4 (IPv4 Based Address): Son aquellas direcciones que se obtienen añadiendo un prefijo de 96 ceros a una dirección IP versión 4 (10.0.0.1 pasaría a ser en la versión 6 **::10.0.0.1**).

4. Direcciones locales reservadas (Site Local Address): Estas direcciones son reservadas para intranets y no son válidas por Internet y tan sólo sirven para que una organización pueda crear sus redes basada en un esquema TCP/IP sin la necesidad de estar conectados a Internet (en la versión 4 de IP, existen diferentes clases reservadas para este mismo fin, como por ejemplo 192.168.XXX.YYY)

5. Direcciones de inicialización locales reservadas (Link Local Address): Son direcciones que pueden utilizar los ordenadores conectados a una misma red local mientras se inicializa y no tiene asignada una dirección IP. Se diferencia de la dirección no especificada (0:0:0:0:0:0:0:0) en que pueden circular por la red, permitiendo por ejemplo obtener el sistema operativo de un servidor en la misma red.

1.7.2 Direcciones Anycast IPv6

Una dirección anycast identifica múltiples interfaces. Con una topología de enrutadores adecuada a los paquetes destinados a una dirección anycast se entregaran a una sola interfaz. Si una dirección multicast define una comunicación Uno a Muchos, no tienen un espacio propio dentro del direccionamiento IPv6 utilizan el mismo espacio que las direcciones unicast (es decir, no se puede diferenciar entre direcciones unicast y anycast). El ámbito de las direcciones anycast se equipará con el unicast, así pues, pueden existir direcciones anycast de ámbito de sitio, de enlace o global. También se puede remarcar que este tipo de direcciones solo pueden usarse como dirección de destino, jamás como fuente. Existe una dirección anycast requerida; su sintaxis es equivalente al prefijo que se especifica el enlace correspondiente de la dirección unicast, siendo el indicador de la interfaz igual a cero:

n bits	**128-n bits**
Prefijo de subred	**0**

Figura 8 Estructura de direcciones Anycast.

Todos los enrutadores soportan estas direcciones para las subredes a las que están conectados, Los paquetes enviados a la “dirección anycast del enrutador de la subred”, serán enviados a un enrutador de la subred. La utilidad de estas direcciones es para implementar los siguientes mecanismos:

- Comunicación con el servidor más cercano. Estas direcciones permiten que un cliente pueda comunicarse con el servidor del grupo.
- Descubrimiento de servicios, al configurar un nodo con IPv6, no falta especificarle la dirección del servidor DNS, Proxy etc. Podría existir una dirección anycast que identificara esos servicios
- Movilidad: los nodos que tienen que comunicarse con el enrutador, del conjunto disponible.

1.7.3 Direcciones multicast

Una dirección multicast en IPV6, puede definirse como un identificador para un grupo de nodos. Un nodo puede pertenecer a uno o varios grupos multicast.

8	4	4	112 bits
11111111	**00T**	**Ámbito**	**Formato direcciones multicast**

---Los primeros 8 bits indican que se trata de una dirección del bit

---La letra "T" indica:

T=0; indica descripción permanente, asignada por la autorización de global de Internet.

T=1; indica una dirección temporal.

Figura 9 Formato de direcciones multicast.

Los bits ámbitos tienen las siguientes descripciones:

Valor	Descripción
0	Reservado
1	Ámbito local de nodo
2	Ámbito local de enlace
3	No asignado
4	No asignado
5	Amito local de sitio
6	No asignado
7	No asignado
8	Ámbito local de organización
9	No asignado
A	No asignado
B	No asignado
C	No asignado
D	No asignado
E	Ámbito Global
F	Reservado

Tabla 3 Descripción de los bits en el ámbito de multicast.

1.7.4 Representación de las direcciones IPv6

IPv4 tiene 32 bits (4 octetos) de longitud cada octeto es representado por un entero no asignado, y los cuatro octetos se escriben como cuatro números decimales, separados por tres puntos cada bloque, por ejemplo: 130.192.1.143.

Para las direcciones IPv6 se toma la sintaxis a partir de la RFC 1884[13] que recomienda 128 bits (16 octetos), cada bloque de 16 bits se convierte a un número hexadecimal de 4 dígitos separado por un signo de dos puntos, como por ejemplo:

a) **FEDC:BA98:7654:3210:FEDC:BA98:7654:3210**
b) **1080:0000:0000:0000:0008:0800:200C:417A**
c) **0000:0000:0000:0000:0000:0000:0A00:0001**

Dado que, por el direccionamiento que se ha definido, pueden existir largas cadenas de bits "cero" (ver ejemplo anterior b y c), se permite la escritura de su abreviación , mediante el uso de "**::**", que representa múltiples grupos consecutivos de 16 bits "cero".

Este símbolo sólo puede aparecer una vez en la dirección IPv6. Por ejemplo podemos escribir los ejemplos b y c de la siguiente manera:

b) 1080::8:800:200C:417A
c) ::A00:1

Si se consideran las direcciones para el caso de multicast, loopback, o dirección no especificada, se puede representar de forma extendida o con el método abreviado:

Forma extendida:

FF01:0:0:0:0:0:0:43 dirección multicast.
0:0:0:0:0:0:0:1 dirección para loopback
0:0:0:0:0:0:0:0 dirección no especificada

Forma abreviada:

FF01::43 dirección multicast
::1 dirección para loopback
:: Dirección no especificada

[13] *RFC 1884, Arquitectura de Direccionamiento para el IP Versión 6*

En resumen, se pueden seguir los siguientes lineamientos para poder manejar las direcciones IPv6:

a). Supresión de los ceros redundantes situados a la izquierda.

1080:0000:0000:0000:0008:0800:200C:417C **Simplificación** 1080:0:0:0:8:800:200C:417C
1080:0:0:0:8:800:200C:417C ⟶ 1080::8:800:200C417C

b) Simplificación de los ceros consecutivos mediante el uso del prefijo '**::**'.

0:0:0:BC98:6564:0:0:0 **Simplificación** ::BC98:6564:0:0:0
⟶ :BC98:6564::
(**Incorrecto**) ::BC98:6564::

c) Para las direcciones IP versión 6 obtenidas añadiendo 96 ceros a la dirección IP versión 4 (10.0.0.1 -> 0:0:0:0:0:0:A00:1) se permitirá el uso de la notación decimal (::10.0.0.1).

0:0:0:0:0:0:0A00:001 **Notación decimal** ::10.0.0.1
::A00:1 ⟶

1.7.4.1 Prefijos

En IPv6 los prefijos de las direcciones son representados por la siguiente notación:

Dirección IPv6 / longitud del prefijo.

La especificación de un prefijo de direccionamiento en la versión 6 se realizará mediante la forma dirección IPv6/prefijo. A continuación se muestran algunos ejemplos:

1. Si tenemos el prefijo de 40 bits FEDC:BA98:76 en la dirección FEDC:BA98:7600::1 se especificará como **FEDC:BA98:7600::1/40**.

2. Para indicar una subred con el prefijo de 80 bits, se suele seguir la siguiente notación:
1080:0:0:0:8::/80
Se debe de notar en este caso los ceros que se encuentran en el centro no son eliminados por que la notación **::** se ha utilizado al final de la dirección.

3. Por ejemplo los 60-bit del prefijo:
12AB00000000CD3
Tiene las siguientes representaciones:
12AB:0000:0000:CD30:0000:0000:0000:0000/60
12AB::CD30:0:0:0:0/60
12AB:0:0:CD30::/60

1.7.4.2 Direcciones Reservadas

En IPv6 también existen direcciones que son reservadas, en la tabla 4 se detallan las direcciones IP que no se deben de utilizar, por que se encuentran reservadas, la longitud del prefijo, así como la descripción correspondiente:

Dirección IPv6	Longitud del Prefijo (Bits)	Descripción	Comentarios
::	128 bits	Sin especificar	Como 0.0.0.0 en Pv4.
::1	128 bits	Dirección de bucle local (loopback)	Como las 127.0.0.1 en IPv4
::00:xx:xx:xx:xx	96 bits	Direcciones IPv6 compatibles con IPv4	Los 32 bits más bajos contienen una dirección IPv4. También se denominan direcciones "empotradas."
::ff:xx:xx:xx:xx	96 bits	Direcciones IPv6 mapeadas a IPv4	Los 32 bits más bajos contienen una dirección IPv4. Se usan para representar direcciones IPv4 mediante direcciones IPv6.
fe80:: - feb::	10 bits	Direcciones link-local	Equivalentes a la dirección de loopback de IPv4.
fec0:: - fef::	10 bits	Direcciones site-local	Equivalentes al direccionamiento privado de IPv4
ff::	8 bits	multicast	
001 (base 2)	3 bits	direcciones unicast globales	Todas las direcciones IPv6 globales se asignan a partir de este espacio. Los primeros tres bits siempre son "001".

Tabla 4 Direcciones IPv6 reservadas

1.8 Configurando IPv6 en los Sistemas Operativos.

Para utilizar IPv6 en el ordenador se necesita tener instalado como protocolo de red el software IP versión 6, el cual está disponible para prácticamente todos los sistemas operativos. En el caso de Windows XP y Windows 2003, el software viene incorporado con el sistema operativo, aunque es necesario instalarlo ya que no viene instalado como protocolo de red por defecto. En el capítulo 11 se presentan las instrucciones adecuadas para instalar, configurar y probar el protocolo de IPv6.

En la tabla 5 se muestran los diferentes sistemas operativos, y las sentencias para ejecutar el protocolo de IPv6. En el siguiente apartado, se muestra unos ejemplos para establecer IPv6 en los sistemas operativos Windows XP y Linux distribución Red Hat.

Sistema operativo	Descripción
Solaris	***/etc/hostname6.ifname***
Red Hat	***NETWORKING_IPV6="yes"* to */etc/sysconfig/network*.**
WinXP	IPv6 install
Win2003	**netsh interface** , o tambien puede utilizar **IPv6 install**
FreeBSD	***IPv6_enable="YES"* to */etc/rc.conf*.**
Mac OS X	Habilitar por defecto ***/etc/hostconfig***

Tabla 5. Formas de ejecutar el protocolo IPv6, en distintos sistemas Operativos.

Capítulo

2

Seguridad en IPv6

Temas:

- Características de seguridad en IPv6.
- Aspectos técnidos de la seguridad en Internet.
- Requerimientos básicos de seguridad.
- Seguridad en Internet y algunas soluciones.
- Protección en la capa de transporte.
- Aplicaciones de seguridad.
- Diferencias de seguridad entre IPv4 e IPv6.

2. Seguridad en IPv6

En la actualidad, la seguridad informática ha adquirido gran auge, dadas las cambiantes condiciones y las nuevas plataformas de computación disponibles la posibilidad de interconectarse a través de la red, ha abierto nuevos horizontes que permiten explorar más allá de las fronteras de la organización. Esta situación ha llevado a la aparición de nuevas amenazas en los sistemas computarizados.

Originalmente las redes fueron diseñadas para intercambiar información entre los investigadores. Hoy en día, Internet ha experimentado un aumento de número de usuarios e interacciones. Estas interacciones requieren de un alto nivel de seguridad, desde la correcta identificación de los participantes, hasta métodos de encriptación.

Internet ha sobrepasado el crecimiento, y los mecanismos de seguridad que se requieren en las aplicaciones, algunos no son parte del diseño original de IP. El desarrollo de una nueva versión del protocolo IP, ha ofrecido una oportunidad de cambiar e introducir algunos mecanismos básicos de seguridad en los niveles de redes que pueden estar disponibles para todas las aplicaciones que se encuentren en las distintas capas del modelo OSI [14]o TCP/IP.

El presente capítulo se enfoca en el tema de seguridad, así como los elementos necesarios para construir una red segura, también se abordan los diferentes ataques, vulnerabilidades que pueden suceder en la implementación de una red, y algunas soluciones ante estos problemas. Los temas que se presentan a continuación, son esenciales, para los capítulos posteriores, debido a que a lo largo de ésta sección encontrara temas básicos de seguridad, que son integrados en IPv6.

[14] *Modelo de referencia de Interconexión de Sistemas Abiertos (OSI, Open System Interconection) fue el modelo de red descriptivo creado por ISO. Proporcionó a los fabricantes un conjunto de estándares que aseguraron una mayor compatibilidad e interoperabilidad entre los distintos tipos de tecnología de red producidos por las empresas a nivel mundial. Ver anexo 1.*

2.1 Características de seguridad en IPv6

El protocolo IPv6, incorpora el protocolo de seguridad de Internet (IPSec), que protege los datos IPv6 cuando se envían a través de la red. IPSec es un conjunto de estándares de Internet que utilizan servicios de seguridad criptográficos para proporcionar las características siguientes:

- Confidencialidad: el tráfico de IPSec está cifrado. El tráfico de IPSec capturado no se puede descifrar si no se conoce la clave de cifrado.

- Autenticación: el tráfico de IPSec está firmado digitalmente con la clave de cifrado compartida de manera que el destinatario pueda comprobar que lo envió el interlocutor IPSec.

- Integridad de los datos: El tráfico de IPSec contiene una suma de comprobación criptográfica que incorpora la clave de cifrado. El destinatario puede comprobar que el paquete no se ha modificado durante la transmisión.

2.2 Aspectos técnicos de la seguridad en Internet

Durante los últimos años, la economía ha crecido aceleradamente y se ha experimentado un aumento en las empresas que implementan comercio electrónico.
El Comercio Electrónico ó e-commerce es una metodología moderna en el proceso de comercialización, ayudada por la tecnología de punta como una nueva maniobra para el desarrollo de una mejor ventaja competitiva para las empresas de todo el mundo, en este sentido las tiendas virtuales tienen un lugar importante, como herramienta de comercialización y expansión geográfica de los negocios, son un elemento fundamental para la globalización actual de la economía.

Dichas aplicaciones comerciales, requieren de redes que brinden servicios de multimedia como por ejemplo integración de voz, video, y rapidez en el tráfico de datos, para poder brindar facilidades y comodidades a los consumidores. Este tipo de aplicaciones, conecta a una gran cantidad de usuarios, por lo tanto las redes de hoy en día presentan cierta vulnerabilidad ante la presencia de ataques y amenazas.

Si se desea mantener la supervivencia y bienestar de muchos negocios, se necesita considerar el aspecto de la seguridad; para que exista una libre y limpia transacción de los datos, y un acceso óptimo hacia ellos. La seguridad marcha en paralelo a la vanguardia de muchas empresas. A continuación se mencionan algunas razones que justifican la importancia de la seguridad:

- La seguridad es un requerimiento indispensable en el comercio electrónico, debido a la necesidad para mantener los datos privados. Se necesita proteger a la información que se transporta a través de las redes públicas.

- La implementación de medidas de seguridad en ambientes que presentan riesgos y vulnerabilidad en la comunicación de los datos es importante por que los negocios se desarrollan a través de la red y necesitan de una interconexión entre las redes públicas.

En el transcurso de esta sección se abordan diferentes aspectos básicos de seguridad que se deben de tomar en cuenta:

2.2.1. Diseño

El diseño de la red es muy importante, en años anteriores, la mayoría de los modelos de redes no consideraban la seguridad como un elemento importante e indispensable, por que el área de la red era pequeña y se desarrollaba solo en ambientes privados; a continuación se presentan los dos modelos de redes, como era la infraestructura hace unos años y como es ahora.

Diseño que se utilizaba hace unos años: considerado como redes cercanas, este tipo de modelo, estaba diseñado para implementarse en un ambiente corporativo, y provee conectividad solo para sitios localmente cercanos, planeado para empresas que no necesitaban de una conexión a una red publica. En años anteriores, la mayoría de las empresas seguían este modelo, por que no existía la necesidad de una comunicación externa, en esos años fueron consideradas seguras. (Ver figura 15 y 16).

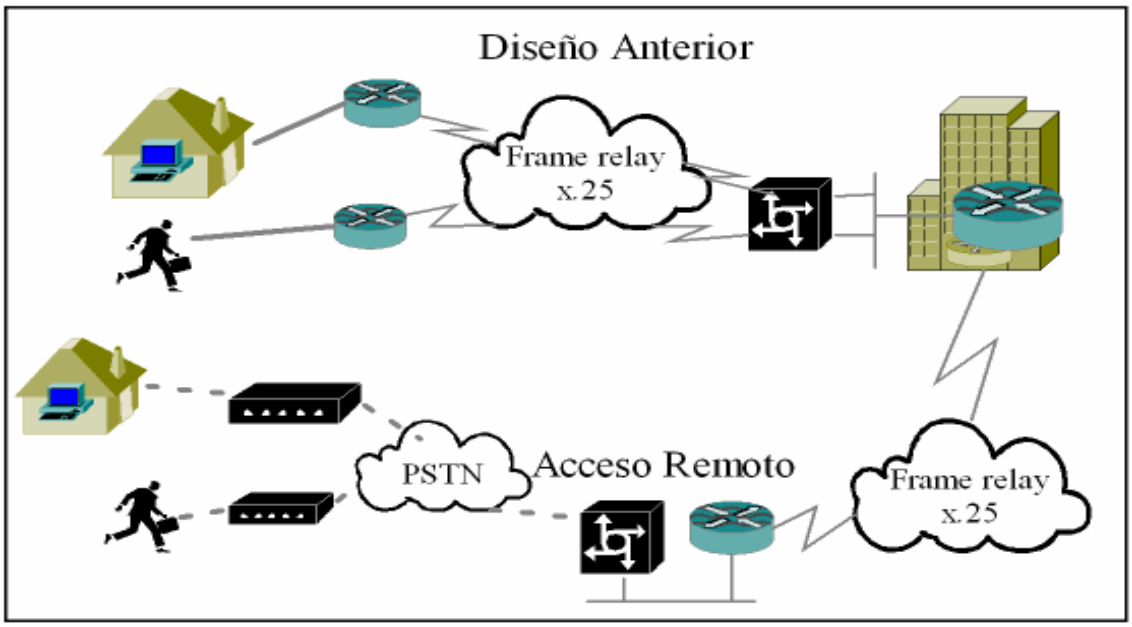


Figura 15 Modelo de diseño de red que se utilizaba hace unos años atrás.

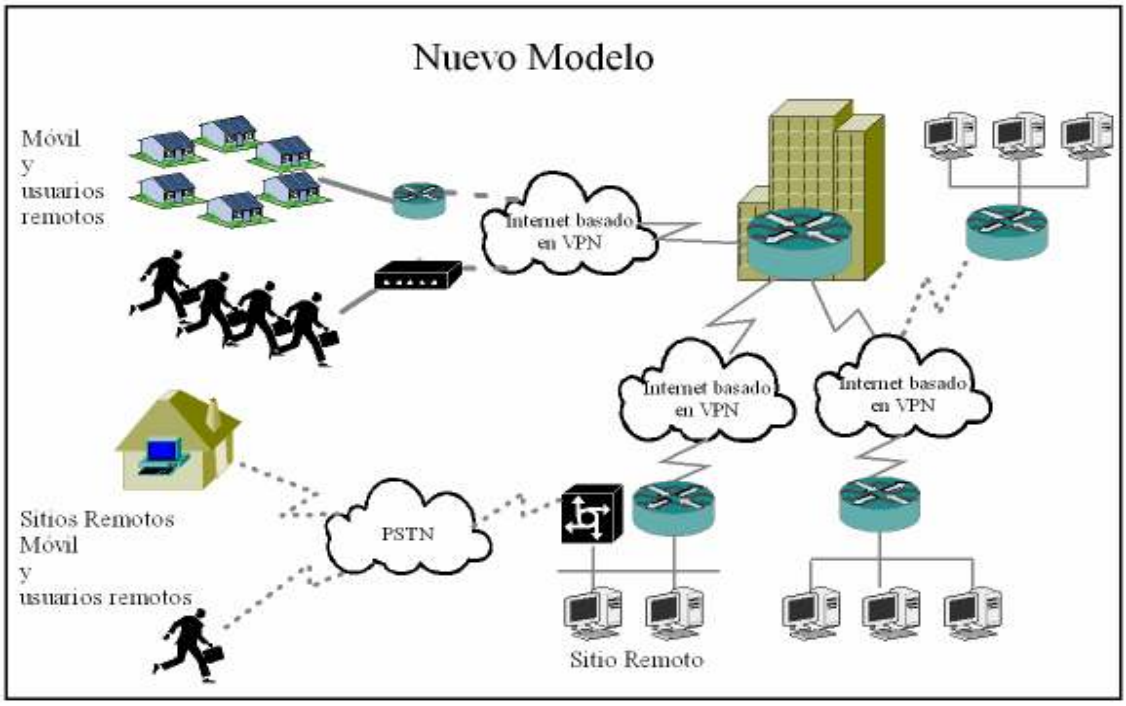


Figura 16 Modelo de diseño de red que se utiliza actualmente.

2.2.2. Elementos básicos de seguridad

Para poder garantizar la confiabilidad del transporte de los datos y la integridad de la información se necesitan de herramientas básicas que se utilizan como medidas de seguridad entre ellas se pueden mencionar (Ver figura 17):

a) Monitoreo: para poder garantizar que la red permanezca segura, es importante monitorear el estado de la ejecución de la seguridad. Con el monitoreo se puede detectar las partes vulnerables de la red y se identifican las áreas que presentan mayor debilidad. Cuando se utilizan soluciones de monitoreo, las identidades pueden obtener mejor visibilidad del comportamiento del flujo de los datos de la red.

b) Comprobación: Comprobar la seguridad es importante así como monitorearla. Si no se implementan pruebas no se pueden conocer la existencia de los nuevos ataques.

c) Incremento en la seguridad de los datos: el monitoreo y las pruebas que se realizan, nos indican cuales son los datos que se requieren de un incremento en su seguridad.

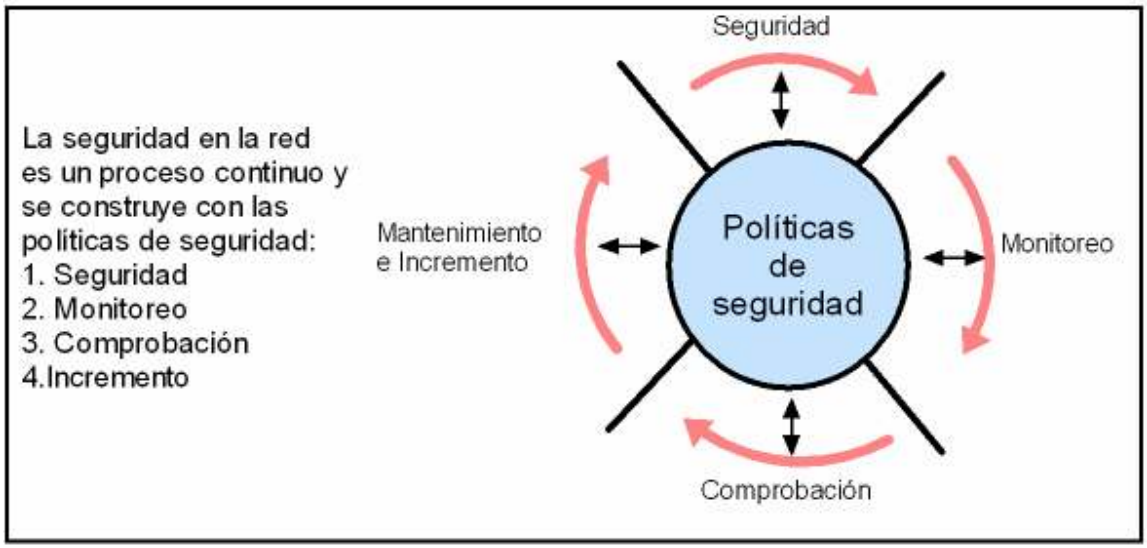


Figura 17 Elementos básicos de seguridad

2.2.3. Políticas de seguridad

En el proceso del diseño de las redes, se necesita aplicar políticas de seguridad, las cuales constituyen el primer paso que se debe de realizar para poder ofrecer un ambiente seguro. Cuando se adoptan las políticas de seguridad, las instituciones o compañías deben de considerarlas como parte de los procesos normales en las operaciones de la Red.

De acuerdo a la RFC 2196 [15], una política de seguridad (PSI[16]) es una forma de comunicarse con los usuarios y los gerentes. Las PSI establecen el canal formal de actuación del personal, en relación con los recursos y servicios informáticos, importantes

[15] *Guía para desarrollar políticas y procedimientos de seguridad para sitios que tienen sistemas en red.*
[16] *Política de Seguridad Interna*

de la organización. No se trata de una descripción técnica de mecanismos de seguridad, ni de una expresión legal que involucre sanciones a conductas de los empleados. Es más bien una descripción de lo que se desea proteger y el por qué de ello. Cada PSI es consciente y vigilante del personal por el uso y limitaciones de los recursos y servicios informáticos críticos de la compañía.

Una PSI debe orientar a las decisiones que se toman en relación con la seguridad. Por tanto, se requiere de una disposición por parte de cada uno de los miembros de la empresa para lograr una visión conjunta de lo que se considera importante. Las PSI deben considerar los siguientes elementos:

a) Alcance de las políticas incluyendo facilidades, sistemas y personal sobre la cual aplica. Es una invitación de la organización a cada uno de sus miembros a reconocer la información como uno de sus principales activos así como, un motor de intercambio y desarrollo en el ámbito de sus negocios.

b) Objetivos de la política y descripción clara de los elementos involucrados en su definición.

c) Responsabilidades por cada uno de los servicios y recursos informáticos de todos los niveles de la organización.
d) Requerimientos mínimos para de la configuración de la seguridad de los diferentes sistemas que pretenden cubrir los alcances de las políticas.

e) Definición de violaciones y de las consecuencias del no cumplimiento de la política

f) Responsabilidades de los usuarios con respecto a la información a la que se tiene acceso. Las PSI deben ofrecer explicaciones comprensibles acerca del porque se deben de tomar ciertas decisiones y advertencias.

2.2.4. Amenazas y ataques en la red.

Sin una apropiada protección, cualquier parte de la red puede ser sensible a ataques o actividades no autorizadas. Los enrutadores, switches y hosts pueden ser intervenidos por hackers [17] profesionales, compañías que son competencia o empleados dentro de una determinada institución. Sin medidas de seguridad, tanto las redes públicas como las privadas están expuestas a la observación y el acceso no autorizados.

Las amenazas a las que están expuestas las redes, se pueden clasificar en cuatro categorías generales, las cuales son las siguientes:

a) No estructuradas: estas amenazas consisten en que cualquier hacker utiliza varias herramientas comunes, como scripts, passowords, los cuales son generadores de números de tarjetas de créditos. En esta categoría, la mayoría de los hackers están interesados en el reto intelectual de descifrar medidas preventivas que en crear una destrucción en la red de una corporación.

b) Estructuradas: en ésta categoría las amenazas son creadas por hackers que tienen una intención de destrucción más alta, y el nivel técnico que poseen también es alto. Tales hackers actúan solos o en grupos pequeños para entender bien el desarrollo de la red, y emplean sofisticadas técnicas para penetrar fácilmente en las compañías. Estos grupos suelen estar involucrados en la mayoría de los fraudes y en casos de robos reportados.

c) Externas: estas clasificación; se originan de los tipos de amenazas estructuradas y no estructuradas, se producen desde una fuente externa. Dichas amenazas pueden tener intenciones destructivas, en algunas ocasiones son el resultado de errores externos que se convierten en amenazas y afectan a la red.

[17] ***Hacker***, *este término es utilizado para referirse a un experto) en varias o alguna rama técnica relacionada con las Tecnologías de la Información y las Telecomunicaciones.*

d) Internas: por lo general este tipo de amenazas surgen dentro de la estructura interna de la empresa, es decir a través de los empleados, que posiblemente manifiesten descontento ante la empresa, o simplemente se genere un descuido en la red.

Aunque las amenazas internas son menores que las externas, las medidas de seguridad deben de estar accesibles para reducir la vulnerabilidad de la red.

2.2.5 Tipos de ataques a redes

Un intruso ataca a las redes o los sistemas para obtener los datos e información, obtienen el acceso y luego ascienden en los accesos de los privilegios. A través de los ataques los intrusos pretenden descubrir y seguir los pasos de los sistemas de los servicios y vulnerabilidades de una empresa o institución.

Si no se emplean medidas de seguridad, ni se aplican controles, los datos pueden ser objeto de un ataque. Algunos ataques son pasivos, en el sentido de que sólo se observa la información. Otros ataques son activos y modifican la información con intención de dañar o destruir los datos o la propia red. Cuando no se tiene un plan de seguridad, las redes y los datos son vulnerables a todos los siguientes tipos de ataques:

1. Espionaje:

En general, la mayoría de las comunicaciones por red están dadas en formato de texto simple (sin cifrar), lo que permite al atacante que haya logrado el acceso a las rutas de datos de una red, observar e interpretar (leer) el tráfico. El espionaje de las comunicaciones por parte de un atacante se conoce como husmear (conocido como sniff). La capacidad de los espías para observar la red suele ser el mayor problema de seguridad que afrontan los administradores de redes de las compañías. Sin servicios de cifrados eficaces basados en criptografía, los datos que atraviesan la red, pueden ser observados fácilmente por terceros.

2. Modificación de datos

Cuando un atacante ha leído los datos, a menudo el siguiente paso lógico consiste en modificarlos. Un atacante puede modificar los datos de un paquete sin que el remitente ni el receptor se den cuentan o sospechen de lo que ha ocurrido. Incluso cuando no se requiera confidencialidad en todas las comunicaciones, y no se desea que los mensajes se modifiquen en su camino. Por ejemplo, si se intercambia solicitudes de una compra, y no se desea que se modifique la información relativa de los artículos, los importes ni la facturación.

3. Suplantación de identidad (direcciones IP ficticias)

La mayoría de las redes y sistemas operativos utilizan la dirección IP para identificar un equipo como válido en una red. En algunos casos, es posible utilizar una dirección IP falsa. Esta práctica se conoce como suplantación (conocido como spoofing). Un atacante podría utilizar programas especiales para construir paquetes IP que parezcan provenir de direcciones válidas dentro de la intranet de una organización. Una vez obtenido el acceso a la red con una dirección IP válida, el atacante podrá modificar, desviar o eliminar los datos.

4. Ataques basados en contraseñas

Un procedimiento común en la mayoría de los sistemas operativos y planes de seguridad de redes es el control de acceso basado en contraseñas. El acceso tanto de un equipo como de los recursos de la red se determina por un nombre de usuario y una contraseña. Históricamente, muchas versiones de componentes de sistemas operativos no siempre protegían la información de identidad cuando ésta pasaba por la red para su validación. Esto puede permitir a un espía detectar un nombre de usuario y una contraseña que sean válidas, y utilizarlas para lograr acceso a la red haciéndose pasar por un usuario autorizado.

Cuando un atacante encuentra una cuenta de usuario válida y la utiliza para el acceso, obtendrá los mismos derechos que el usuario real. Por ejemplo, si el usuario tiene derechos administrativos, el atacante puede crear cuentas adicionales para tener acceso posteriormente. Una vez obtenido el acceso a una red con una cuenta válida, el atacante puede hacer lo siguiente:

- Obtener listas de nombres de usuarios y equipos válidos e información de la red.
- Modificar las configuraciones de los servidores y de la red, incluyendo los controles de acceso y las tablas de enrutamiento.
- Modificar, desviar o eliminar datos.

5. Ataque de negociación (Denial of Service DoS)

A diferencia de un ataque basado en contraseñas, el ataque de rechazo de servicio impide el uso normal de un equipo o de una red por parte de los usuarios autorizados. Una vez obtenido el acceso a una red, el atacante puede hacer lo siguiente:

- Distraer al personal de sistemas de información para que no detecte inmediatamente la intrusión. Este método indica al atacante la oportunidad de llevar a cabo ataques adicionales.
- Enviar datos no válidos a aplicaciones o servicios de red para provocar su cierre o su funcionamiento de forma anormal.
- Generar tráfico de datos, masivamente hasta provocar el colapso de un equipo o de toda la red.
- Bloquear el tráfico de datos, lo que hace perder el acceso a los recursos de la red por parte de los usuarios autorizados.

6. Ataque por Man in the Middle

Como su nombre indica, un ataque por usuario interpuesto se produce cuando alguien situado entre dos usuarios que se están comunicando observa activamente, captura y controla la comunicación sin que los usuarios lo adviertan. Por ejemplo, un atacante puede negociar claves de cifrado con ambos usuarios. A continuación, cada usuario enviará datos cifrados al atacante, quien podrá descifrarlos. Cuando los equipos se comunican en niveles bajos de la capa de red, quizás no puedan determinar con qué equipos están intercambiando datos.

7. Ataque de clave comprometida

Una clave es un código o un número secreto necesario para cifrar, descifrar o validar información protegida. Averiguar una clave es un proceso difícil y requiere grandes recursos por parte del atacante, pero no deja de ser posible. Cuando un atacante averigua una clave, ésta se denomina clave comprometida. El atacante puede utilizar la clave comprometida para obtener acceso a una comunicación protegida sin que el remitente ni el receptor lo perciban. La clave comprometida permite al atacante descifrar o modificar los datos. El atacante también puede intentar utilizar la clave comprometida para calcular otras claves que podrían suponer el acceso a otras comunicaciones protegidas.

8. Ataque en la capa de aplicación

Los ataques en la capa de aplicación se dirigen a los servidores de aplicaciones e intentan provocar errores en su sistema operativo o en sus aplicaciones. De este modo el atacante puede llegar a eludir los controles de acceso normales. El atacante aprovecha esta situación para obtener el control de una aplicación, sistema o red, con lo que podrá hacer lo siguiente:

- Leer, agregar, eliminar, modificar los datos o un sistema operativo.
- Introducir un virus que utilice los equipos y las aplicaciones de software para propagarse por toda la red.
- Introducir un programa husmeador que analice la red y obtenga información que pueda utilizarse para hacer que la red deje de responder o que resulte dañada.
- Cerrar aplicaciones de datos o sistemas operativos de forma anormal.
- Deshabilitar otros controles de seguridad para posibilitar futuros ataques.

2.3 Requerimientos básicos de seguridad.

De acuerdo a las descripciones de los diferentes tipos de amenazas y ataques que pueden ejecutarse en una red, es necesario conocer e implementar ciertos requisitos básicos pero importantes en la arquitectura de una red, como lo son los siguientes:

2.3.1 Confidencialidad

Consiste en proteger la información contra la lectura no autorizada explícitamente. Incluye no sólo la protección de la información en su totalidad, sino que también las piezas individuales que pueden ser utilizadas para inferir otros elementos de información confidencial. (Ver figura 18).

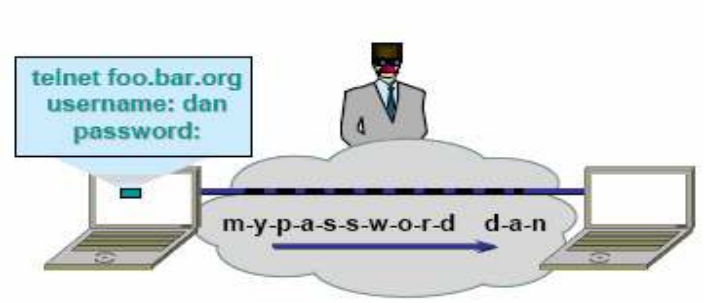


Figura 18 La comunicación entre dos nodos debe de ser confiable.

2.3.2 Integridad

Es necesario proteger la información contra las modificaciones que se pueden dar sin el permiso o consentimiento del propietario. La información que debe ser protegida no solo es la que se encuentra almacenada directamente en los sistemas de cómputo; sino que también se deben de considerar elementos menos obvios como respaldos, documentación, registros de contabilidad del sistema, tránsito en una red, etc. (Ver figura).

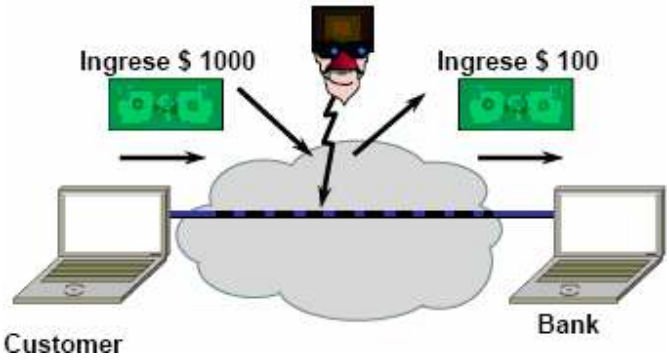


Figura 19 La integridad de las transacciones es un requerimiento de seguridad.

Esta protección se realiza para mantener la integridad de los datos, ayuda a prevenir la pérdida de ellos ante las situaciones siguientes:

- Causadas por errores de hardware y /o software.
- Causadas de forma intencional
- Causadas de forma accidental
- Cuando se trabaja con una red, se debe comprobar que los datos no fueron modificados durante su transferencia.

2.3.3 Autenticidad

En cuanto a telecomunicaciones se refiere, la autenticidad garantiza que quien dice ser "X" es realmente "X". Es decir, se deben implementar mecanismos para verificar quién está enviando la información. (Ver figura 20).

Figura 20 La autenticidad brinda validación en la identificación.

2.3.4 No repudio

Ni el origen ni el destino en un mensaje deben poder negar la transmisión. Quien envía el mensaje puede probar que, en efecto, el mensaje fue enviado y viceversa. (Ver figura 21).

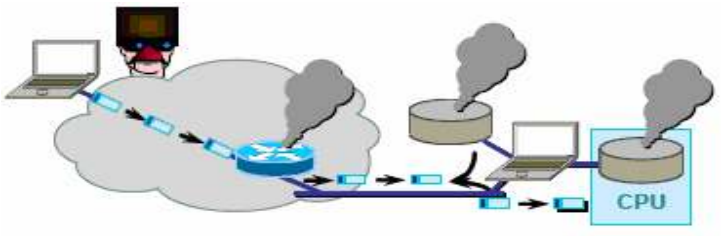


Figura 21 Es irrevocable, no se puede rechazar por su propietario.

2.3.5 Encriptación

Encriptación es el proceso mediante el cual cierta información o "texto plano" es cifrado de forma que el resultado sea ilegible; a menos que se conozcan los datos necesarios para su interpretación. Es una medida de seguridad utilizada para que al momento de almacenar o transmitir información sensible ésta no pueda ser obtenida con facilidad por terceros. Opcionalmente puede existir además un proceso de desencriptación a través del cuál, la información puede ser interpretada de nuevo a su estado original, aunque existen métodos de encriptación que no pueden ser revertidos.

La encriptación hace uso de diversas fórmulas matemáticas con el propósito de transformar el texto plano en un *criptograma* el cual, es un conjunto de caracteres que a simple vista no tiene ningún sentido para el lector. La mayoría de los métodos de encriptación utilizan una clave como parámetro variable en las mencionadas fórmulas matemáticas de forma que a pesar de que un intruso las conozca, no le sea posible descifrar el criptograma si no conoce la clave, la cual solo se encuentra en posesión de las personas que pueden tener acceso a la información en cuestión. Algunos métodos utilizan incluso dos claves, una pública para encriptar el texto y una llave privada para desencriptar. El cifrado moderno se divide actualmente en *cifrado de clave privada (o simétrica) y cifrado de clave pública (o asimétrica):*

2.3.5.1 Clave privada (simétrica).

Utiliza una misma clave para cifrar y para descifrar mensajes. Las dos partes que se comunican se deben de poner de acuerdo de antemano sobre la clave a utilizar. Una vez ambas partes tienen acceso a esta clave, el remitente cifra un mensaje usándola, lo envía al destinatario, y éste lo descifra con la misma. (Ver figura 22). En el cifrado de clave privada (simétrica), las claves de cifrado y descifrado son la misma (o bien se deriva de forma directa una de la otra), debiendo mantenerse en secreto dicha clave. Ejemplos de este tipo: DES (Data Encryption Standar), triple DES e IDEA (International Data Encryption Algorithm).

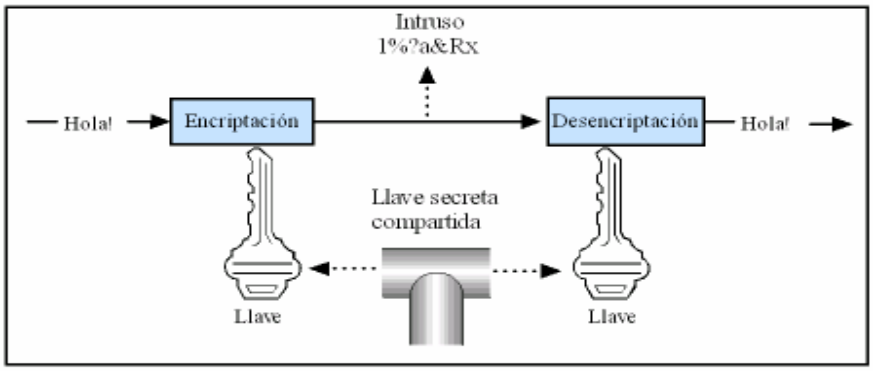


Figura 22 Cifrado Simétrico.

2.3.5.2 Clave pública (asimétrica).

La encriptación asimétrica permite que dos personas puedan enviarse información encriptada, sin necesidad de compartir la llave de encriptación. Se utiliza una llave pública para encriptar el texto y una llave privada para desencriptar. A pesar de que puede parecer extraño que se encripte con la llave pública y desencripte con la privada, el motivo para hacerlo es el siguiente: si alguien necesita que le envíen la información encriptada, deja disponible la llave pública para que quienes le desean enviar algo lo encripten. Nadie puede desencriptar algo con la misma llave pública. El único que puede desencriptar es quien posea la llave privada, quien justamente es el que recibe la información encriptada. (Ver figura 23).

En el cifrado de clave pública (asimétrica), las claves de cifrado y descifrado son independientes, no derivándose una de la otra, por lo cual puede hacerse pública la clave de cifrado siempre que se mantenga en secreto la clave de descifrado. Ejemplos de este tipo: Cifrado RSA (Rivest, Shamir, Adleman). Los algoritmos de encriptación asimétrica más conocidos son:

- **RSA (*Rivest, Shamir, Adleman*)**

Creado en 1978, actualmente es el algoritmo de mayor uso que pertenece a la encriptación asimétrica. Tiene dificultades para encriptar grandes volúmenes de información, por lo que es usado por lo general en conjunto con algoritmos simétricos.

- ***Diffie-Hellman* (& *Merkle*)**

No es precisamente un algoritmo de encriptación sino un algoritmo para generar llaves públicas y privadas en ambientes inseguros.

- ***ECC (Elliptical Curve Cryptography)***

Es un algoritmo que se utiliza poco, pero tiene importancia cuando es necesario encriptar grandes volúmenes de información.

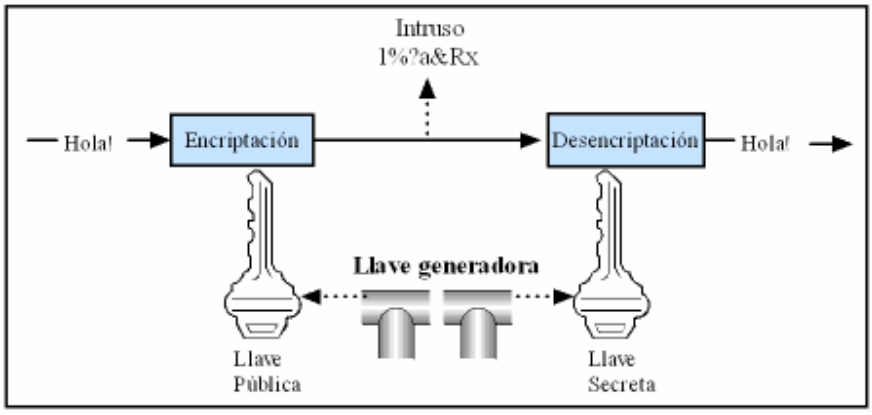


Figura 23 Cifrado Asimétrico.

2.3.5.3 Firma Digital.

La validación de identificación y autenticidad de muchos documentos legales, financieros y de otros tipos se determina por la presencia o ausencia de una firma manuscrita autorizada o bien de una firma digital. La firma digital permite que un nodo pueda enviar un mensaje "firmado" a otro nodo, con las propiedades de autentificación: integridad, autenticidad y no repudio. Por tanto, *la clave de la firma digital está obligada a pedir obligatoriamente un acuse de recibo.* (Ver figura 24).

Requisitos de la Firma Digital:

a) Debe ser fácil de generar.

b) Será irrevocable, no rechazable por su propietario con el acuse de recibo.

c) Será única, sólo posible de generar por su propietario.

d) Será fácil de autenticar o reconocer por su propietario y los usuarios receptores.

e) Debe depender del mensaje (por compendio) y del autor (por certificado).

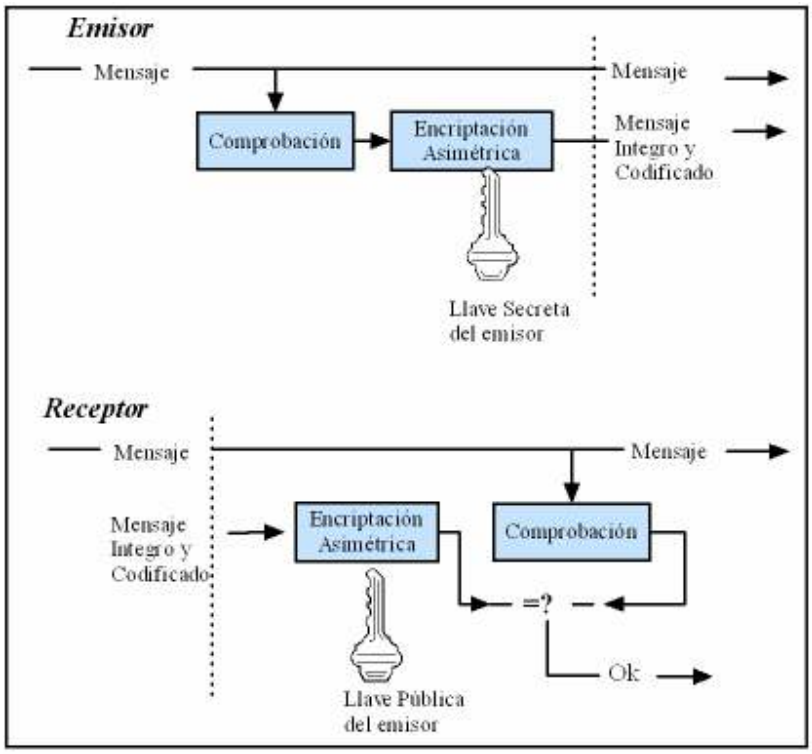


Figura 24 Firma digital.

Al igual que la criptografía, las firmas digitales se dividen en dos grandes grupos:

1. **Firmas de clave secreta o simétrica**
2. **Firmas de clave pública o asimétrica**

1. Firma digital: con clave secreta

Un enfoque de las firmas digitales es tener una autoridad central "X **",** la cual posea el conocimiento total sobre la transmisión y en quien todos los usuarios (emisores y destinatarios) confíen. Cada usuario escoge una clave secreta y la lleva personalmente a la autoridad central "X." Por tanto, sólo el usuario y "X" conocen la clave secreta del usuario. En el caso del usuario A, sería la clave secreta **KA.**

Ejemplo: el algoritmo HMAC (*Hash Message Authentication Code*); que consiste en añadir al final del mensaje, el resumen de éste, pero cifrado con una clave que identifica al usuario. (Ver figura 25).

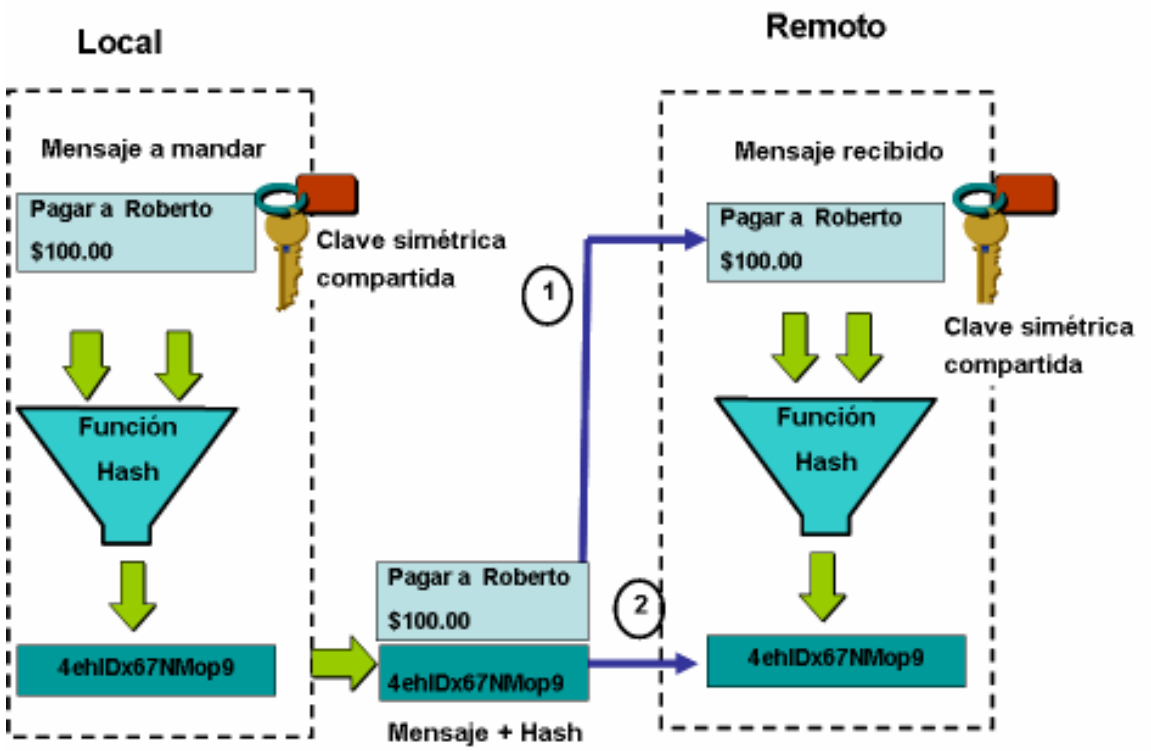


Figura 25 Hash Message Authentication Method (HMAC)

2. Firma digital: con clave pública

Supongamos los algoritmos públicos tal que **E(D(P))=P y D(E(P))=P** (*el RSA tiene esta propiedad por lo que el supuesto es razonable*). **A** puede enviar un mensaje de texto normal firmado, P, a **B** transmitiendo EB (DA(P)), (Ver figura 26), donde:

- DA()es la función de descifrado (privada) de A
- EB() es la función pública de B
- Y por tanto **B**, puede realizar el proceso inverso **EA(DB (EB(DA(P))**
- DB() es la función privada de B
- EA() es la función pública de A

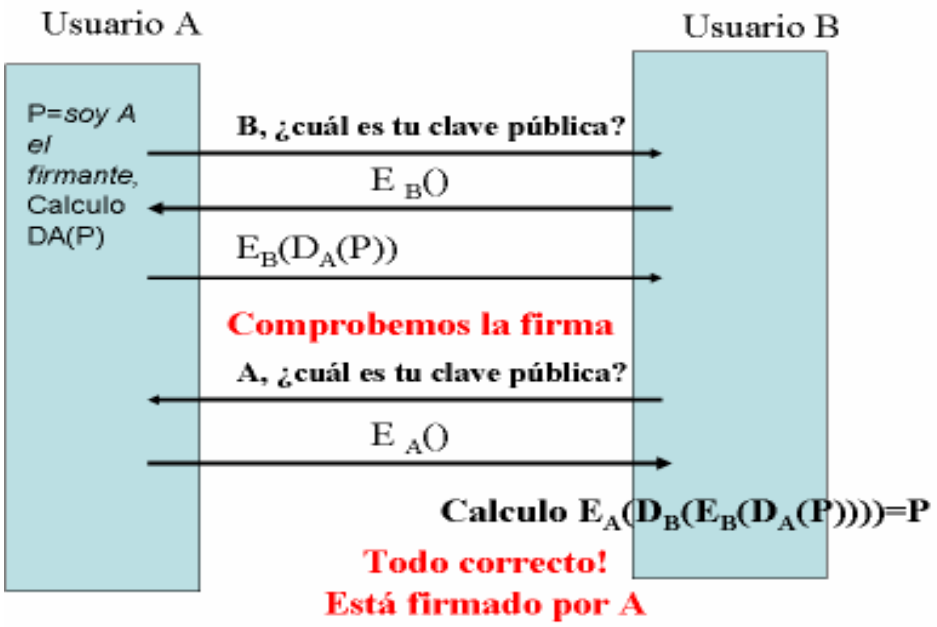


Figura 26 Firma Digital con clave pública.

2.4 Seguridad en Internet y algunas soluciones

Entre las principales razones del éxito de Internet está el hecho de ser una red abierta. Existen millones de usuarios de Internet, y el cálculo estadístico de cuántos individuos tienen acceso a Internet ha perdido ya sentido. Cualquier transacción económica realizada por medios tradicionales es susceptible de ser aprovechada por los intrusos o hackers. Las comunicaciones comerciales realizadas por medios tradicionales, cartas o teléfono, son mucho más fáciles de interceptar que las comunicaciones a través de Internet. Realizar actividades delictivas a través de Internet requiere de conocimientos técnicos sofisticados que no están al alcance de cualquier personal.

Por otra parte, las posibilidades de protección de las comunicaciones electrónicas son mayores que las que permiten los medios tradicionales. Existen programas de ordenador gratuitos que permiten la encriptación de sus mensajes de forma que queda plenamente garantizado que sólo el destinatario podrá entenderlos. Los certificados y firmas electrónicas garantizan la identidad, con mayor garantía que cualquier otro método tradicional. Los sistemas de almacenamiento de datos y su protección frente a accidentes o ataques intencionados son más fáciles, y seguros que las cajas fuertes o cámaras de seguridad.

La protección legal del comercio electrónico ha requerido también la elaboración de nuevas normas. La protección frente a la publicidad indeseada cuyo costo de transmisión recae sobre el consumidor requiere ahora un tratamiento diferente que cuando el coste recaía exclusivamente sobre el anunciante. Los gobiernos de todo el mundo están interesados en promover el desarrollo del comercio electrónico por lo que están impulsando reformas legales y fiscales que permiten y agilicen las transacciones a través de Internet. La seguridad en Internet y las leyes que la protegen, están basadas principalmente en los sistemas de encriptación. Esos sistemas son los que permiten que las informaciones que circulan por Internet sean indescifrables, ininteligibles, para cualquier persona que no sea aquella a la que va destinada.

2.4.1 Soluciones Actuales.

Las soluciones de seguridad han sido agregadas conforme el tiempo. Estas no se consideran totalmente confiables y seguras al acceso a Internet entre ellas se pueden considerar las siguientes:

2.4.2 Filtros de paquetes y Firewalls.

Los motivos básicos para establecer un filtro son la regulación y control del tráfico de de una máquina o red. La regulación se entiende como la decisión del tráfico que se permite y el tipo de tráfico que se prohíbe en función de los orígenes y destinos de los paquetes circulantes. Por ejemplo se puede solicitar todo el tráfico para el servidor Web y sin embargo se puede restringir el envío de correo electrónico, también impide el acceso de una o varias determinadas máquinas de la red, locales o sobre Internet.

El control se entiende como la posibilidad de analizar y manipular las cabeceras de los paquetes para que se adapten a las necesidades. Por ejemplo, el control que permite el enmascaramiento de paquetes, implica que varias máquinas en una red local con direcciones privadas puedan acceder a Internet con una única dirección IP válida. Para hacerlo posible, se tiene que manipular el paquete y sustituir la dirección privada por una dirección pública. La manipulación de paquetes permite otras posibilidades como mantener servidores públicos con direcciones IP privadas que están detrás del Firewall. Los objetivos de la regulación y control se pueden resumir en dos características básicas: *seguridad y rendimiento*. Seguridad porque se puede tomar decisiones sobre el acceso a los servicios y rendimiento por que se puede mejorar ciertas prestaciones de la red y reducir tráfico innecesario.

Se debe de tener en cuenta que la configuración de un firewall no es una tarea común y cotidiana. Una regla de filtrado mal puesta puede inhabilitar las conexiones de red de una máquina. Es necesario conocer en detalle las características de los paquetes IP, los mecanismos para establecer una conexión TCP, el significado de cada protocolo, las características de cada servicio, el esquema de filtrado y otras más.

2.5 Protección en la capa de transporte.

El protocolo SSL (Secure Socket Layer) es un sistema diseñado y propuesto por Netscape Communications Corporation. Se encuentra en la pila OSI entre los niveles de TCP/IP y de los protocolos HTTP, FTP, SMTP, etc. Proporciona sus servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente con un algoritmo de cifrado simétrico, típicamente IDEA, y cifrando la clave de sesión mediante un algoritmo de cifrado de clave pública, típicamente el RSA. La clave de sesión es la que se utiliza para cifrar los datos que vienen del y se dirigen al servidor seguro. Se genera una clave de sesión distinta para cada transacción, lo cual permite que aunque sea revelada en presencia de un atacante en una transacción dada, no sirva para descifrar futuras transacciones. MD5[18] se usa como algoritmo de hash. Proporciona cifrado de datos, autenticación de servidores, integridad de mensajes y, opcionalmente, autenticación de cliente para conexiones TCP/IP.

Cuando el cliente pide al servidor seguro una comunicación segura, el servidor abre un puerto cifrado, gestionado por un software llamado Protocolo SSL Record, situado encima de TCP. Será el software de alto nivel, Protocolo SSL Handshake, quien utilice el Protocolo SSL Record y el puerto abierto para comunicarse de forma segura con el cliente.

El Protocolo SSL Handshake

Durante el protocolo SSL Handshake, el cliente y el servidor intercambian una serie de mensajes para negociar las mejoras de seguridad. Este protocolo sigue las siguientes seis fases:

- La fase Hola, usada para ponerse de acuerdo sobre el conjunto de algoritmos para mantener la intimidad y para la autenticación.
- La fase de intercambio de claves, en la que intercambia información sobre las claves, de modo que al final ambas partes comparten una clave maestra.
- La fase de producción de clave de sesión, que será la usada para cifrar los datos intercambiados.

[18] *Message-Digest Algorithm 5,(Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits.*

- La fase de verificación del servidor, presente sólo cuando se usa RSA como algoritmo de intercambio de claves, y sirve para que el cliente autentique al servidor.
- La fase de autenticación del cliente, en la que el servidor solicita al cliente un certificado X.509[19] (si es necesaria la autenticación de cliente).
- Por último, la fase de fin, que indica que ya se puede comenzar la sesión segura.

El Protocolo SSL Record

El Protocolo SSL Record especifica la forma de encapsular los datos transmitidos y recibidos. La porción de datos del protocolo tiene tres componentes:

- MAC-DATA, el código de autenticación del mensaje.
- ACTUAL-DATA, los datos de aplicación a transmitir.
- PADDING-DATA, los datos requeridos para rellenar el mensaje cuando se usa cifrado en bloque.

2.6 Apicaciones de Seguridad.

Muchas de las nuevas aplicaciones de Internet, permiten implementar sus propios mecanismos de seguridad, en la mayoría de los casos éstos pueden ejecutar los procesos de autenticación y encriptación durante la transmisión, como por ejemplo el protocolo Secure Shell (SSH)[20] , PEM (Privacy Enhanced Mail) y PGP.

Los servicios de seguridad pueden ser agregados a cada enlace de comunicación a lo largo de una trayectoria dada, o pueden ser integrados alrededor de los datos que son enviados, esto independiente de los mecanismos de comunicación. Este enfoque avanzado es frecuentemente llamado seguridad “nodo-a-nodo” (end-to-end).

[19] *Estándar ITU utilizado para la infraestructura de claves pública (Public Key Infrastructure o PKI), especifica formatos estándar para certificados de claves públicas y un algoritmo de validación de ruta de certificación.*
[20] ***SSH (S**ecure **SH**ell) es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red*

Las dos características de este tipo de seguridad son privacidad (donde el recipiente deseado sólo puede leer el mensaje) y la autentificación (en el otro caso, recipiente puede asegurar la identidad del emisor). La capacidad técnica de estas funciones es bien conocida desde hace tiempo, sin embargo, recientemente ha sido sólo aplicada al correo-e de Internet.

Es usual que se cuente con un mecanismo de autentificación de quién origina el mensaje y privacidad para los datos. Además, de proveer un esquema de recepción firmada desde el recipiente. En núcleo de estás capacidades en el uso de la tecnología de llave pública y el uso a gran escala de llaves públicas, lo que requiere un método de certificación que dada una llave pertenece a un usuario dado.

Aunque, se ofrecen servicios parecidos al usuario final, los dos protocolos tienen formatos distintos. Adicionalmente, y esto es importante a los usuarios corporativos, en este caso se cuenta con diversos formatos para los certificados. Lo que significa, que no sólo los usuarios no pueden comunicarse con los que usen otro, además, no pueden compartir los certificados de autenticación. La diferencia entre los dos protocolos es parecida a la diferencia entre los formatos GIF y JPEG, siendo que hacen las mismas cosas, más no su formato entre ellos.

Existen dos propuestas principales para ofrecer los servicios de seguridad que se ha mencionado:

- **PGP:** es una herramienta convencional para el correo electrónico seguro. En sus inicios, PGP utilizaba RSA para las firmas e IDEA para encriptar. IDEA es un cifrador iterativo que procede por bloques de 64 bits con llaves de 128 bits, inmune a criptoanálisis de tipo diferencial. En su especificación estándar, a PGP se le han incorporado triple-DES para el encriptamiento de mensajes, RSA y DSA para firmas electrónicas y MD5, para el cálculo de compendios de mensajes. Los certificados de llaves en PGP contienen dos atributos suplementarios: Confianza y validez. Localmente, la colección de certificados comienza con la propia del usuario.

- **PEM (Privacy Enhanced Mail):** es un estándar oficial de Internet el cual se describe con cuatro RFC's: 1421 al 1424 [21]. PEM cubre el mismo territorio que PGP: privacidad y autentificación para sistemas de correo basados en el RFC 822[22]. Sin embargo, presenta algunas diferencias en enfoque y tecnología. Los mensajes enviados usando PEM son inicialmente convertidos a una forma normalizada, de tal manera que tengan las mismas características sobre el uso de un espacio en blanco, tabuladores, y el uso de retornos de carro y avances de línea. Esta transformación se realiza para eliminar los efectos sobre el agente emisor que transfiere el mensaje evitando que lo modifique o presente tendencia a modificarlo. Sin la normalización, tales modificaciones podrían afectar el mensaje desde que sale hasta que llega a su destinatario. En PGP, el mensaje es encriptado con una llave, al mismo tiempo que protege al mensaje.

- **S/MIME**: proporciona una manera consistente de enviar y recibir datos MIME seguros. Basado en el popular estándar de Internet MIME, S/MIME proporciona los siguiente servicios criptográficos de seguridad: autenticación, integridad y no repudio en origen (utilizando la firma digital) y privacidad y seguridad en los datos (utilizando encriptación). Se puede utilizar en los clientes de correos tradicionales para añadir servicios adicionales de seguridad a los mails enviados y para interpretar los servicios criptográficos de seguridad en los mensajes recibidos. S/MIME no está solo restringido al e-mail y puede ser utilizado por cualquier protocolo de comunicaciones que pueda transportar datos MIME, como por ejemplo HTTP.

[21] *RFC 1421-1424*, "*Privacy Enhancement for Internet Electronic Mail (PEM)"*
[22] *RFC 822, es el formato estándar de Internet para cabeceras de mensajes de correo electrónico.*

2.7 Diferencias de seguridad entre IPv4 e IPv6

Seguridad en IPv4

1. La seguridad es un aspecto adicional

El diseño de TCP/IP (IPv4) no está elaborado para tomar en cuenta aspectos de seguridad. Para poder implementar seguridad en IPv4 se puede seleccionar las siguientes alternativas:

- Seguridad en las aplicaciones.
- Uso de cortafuegos.
- Uso de IPsec.

SSL, Secure Socket Layer, IPsec son extensiones de seguridad diseñadas para IPv4, son algunas de las alternativa que se han tenido que incorporar para brindar seguridad a la Red.

2. IPv4 ofrece un servicio no fiable en dos sentidos:

- No se garantiza la entrega de un datagrama.
- No se comprueba la integridad de los datos del datagrama (sí, en parte, de la cabecera) TCP (sobre todo) se encarga de asegurar la entrega, mediante asentimientos y retransmisiones, pero lo hace extremo a extremo.

Seguridad en IPv6

1. **IPv6 incluye IPsec como parte de su especificación.**

Con la nueva generación de IPv6, la seguridad queda más que contemplada y, lo más importante, incluida en el protocolo. La base de esta seguridad es el protocolo IPSEC, con la diferencia de que los mecanismos de seguridad de este protocolo, ya no son agregados adicionalmente, como ocurre con el IPv4. Debido a que se encuentra agregado en el propio núcleo, lo que le hace más seguro y eficiente de ofrecer un blindaje con más garantías.

2. **IPv6 ofrece un servicio fiable en dos sentidos:**

Los puntos fuertes de la seguridad de IPv6 se basan en la encriptación de los datos que se transportan y los mecanismos de autentificación entre extremos que incorpora y para los que lleva incorporados dentro de los campos, AH, cabecera de autentificación y ESP, encriptación de la carga útil. Ambos mecanismos se complementan con otros elementos de seguridad desarrollados en paralelo, como pueden ser los certificados digitales. La encriptación y autentificación son la base de las VPN, Redes Privadas Virtuales, Virtual Private Networks, que abren muchas posibilidades para el uso de Internet de manera completamente segura.

Capítulo

3

Mecanismos de seguridad de IPv6

Temas:

- Mecanismos de seguridad.
- Términos empleados.
- Asociaciones de seguridad.
- La Cabecera de Autentificación (AH).
- La cabecera de Carga de Seguridad Encapsulada (ESP).

3. Mecanismos de seguridad.

Debido al carácter científico que tuvo en un principio Internet, la seguridad no fue contemplada históricamente en ninguna de las capas que forman la estructura TCP/IP. La tardía reacción de las instituciones encargadas de la creación y modificación de los protocolos de Internet, propició la aparición de diferentes soluciones comerciales para que los usuarios pudieran disfrutar de una seguridad que Internet no proporcionaba.

Aprovechando la necesidad de adaptar los diferentes protocolos al crecimiento de Internet, se optó por introducir una serie de especificaciones para garantizar la seguridad como parte implícita de las nuevas especificaciones de los protocolos. Estas especificaciones se conocen como IP Security o IPSec. Este protocolo está diseñado para proporcionar seguridad ínter-operable, de alta calidad, basada en criptografía, tanto para IPv4 como para IPv6.

La seguridad de IPv6 está heredada de la que proporciona IPSec: especificación de mecanismos de seguridad, que proporcionan autenticación, integridad, control de acceso y confidencialidad, servicios básicos de seguridad. El conjunto de servicios de seguridad ofrecidos por IPsec incluye:

- Control de acceso: previene el uso no autorizado de recursos.
- Integridad sin conexión: detección de modificaciones en un datagrama IP individual.
- Autenticación del origen de los datos.
- Protección anti-replay: una forma de ofrecer integridad, por que detecta la llegada de datagramas IP duplicados.
- Confidencialidad: encriptación.

Las especificaciones IPSec han sido definidas para trabajar en la capa inferior de la pila de protocolos TCP/IP, funcionando por lo tanto en el nivel de datagrama y siendo independientes del resto de protocolos de capas superiores (TCP, UDP). La seguridad en IPSec se proporciona mediante dos aspectos de seguridad:

1. Cabecera de autenticación (Authentication Header, AH). Esta cabecera es la encargada de proporcionar autenticidad a los datos (datagramas) que se reciben en dos aspectos:

- Los datagramas provienen del origen especificado. Se garantiza la autenticidad del origen de los datos (no pueden ser repudiados).
- Los datagramas (y por tanto los datos que contienen) no han sido modificados.

2. Cifrado de seguridad (Encrypted Security Payload, **ESP**). De esta forma se garantiza que tan sólo el destinatario legítimo del datagrama pueda descifrar el contenido del datagrama. La autenticidad y el cifrado de datos requieren que tanto el emisor como el receptor compartan una clave, un algoritmo de cifrado/descifrado y una serie de parámetros que diferencian una comunicación segura de otra. Estos parámetros conforman la asociación de seguridad (Security Association, SA) que permite unir la autenticidad y la seguridad en IPSec.

3.1 Terminología de la seguridad en IPv6

En este capítulo se utilizan términos importantes entre los cuales se pueden mencionar:

- **Control de acceso**: el proceso de prevenir acceso no autorizado a un recurso de red.

- **Autenticación:** la verificación de la identidad de la fuente reclamada de los datos (también conocida como autenticación del origen de los datos).

- **Integridad:** la propiedad de asegurar que los datos son transmitidos desde una fuente o destino sin modificación sin detectar. Integridad sin conexión es un servicio que detecta la modificación de un paquete IP individual, sin importar el orden del paquete en una cadena de datos. Integridad anti-replay (o integridad de secuencia parcial) detecta la llegada de paquetes IP duplicados de una ventana.

- **Confidencialidad:** la protección de los datos de acceso no autorizados.

- **Índice de parámetros de seguridad (SPI)**: un valor de 32 bits que es usado para distinguir entre diferentes Asociaciones de Seguridad (SA's) terminando en el mismo destino y usando el mismo protocolo IPsec.

- **Asociación de seguridad (SA)**: una simple (unidireccional) conexión lógica, creada para propósitos de seguridad. Tanto AH como ESP hacen uso de la SA.

- **Gateway de seguridad**: un sistema que actúa como intermediado entre dos redes. Los hosts o redes en el lado externo del gateway de seguridad son vistos como sistemas no confiables, mientras que los hosts o redes en el lado interno son vistos como sistemas confiables.

3.2 Asociaciones de seguridad

La Asociación de Seguridad (SA) es una conexión lógica simple (o de una vía) que provee servicios de seguridad al tráfico que está siendo cargado sobre esa conexión. Estos servicios de SA pueden ser proveídos a AH o ESP pero no a ambos. Si se desean un AH y un ESP, dos SA son requeridos.

Dos tipos de SA son definidos: modo de transporte y modo de túnel. EL SA en modo de transporte existe entre dos hosts. Para el modo de transporte, el encabezado de protocolo de seguridad (AH o ESP) aparecería después del encabezado IP y otros encabezados de extensión opcionales, pero pueden aparecer antes o después del encabezado de destino, y antes de cualquier encabezado de protocolo de capa más alta como UDP o TCP. Cuando AH es empleado en modo de transporte, la seguridad es proveída para las porciones del encabezado IP y protocolos de capa más alta, Cuando ESP es empleado en modo de transporte, la seguridad es proporcionada solamente para los protocolos de capa más alta.

Una asociación de seguridad individual usa solo un protocolo de seguridad: AH o ESP. Si la política de seguridad dicta habilidades que no son ejecutables con un solo protocolo de seguridad, múltiples SA´s pueden ser usadas para esta implementación. El término paquete de asociación de seguridad es aplicado a esa condición. Las asociaciones de seguridad pueden ser combinadas en paquetes en dos formas:

- Un transporte adyacente
- Tunneling iterado

Con un transporte adyacente, más de un protocolo de seguridad es aplicado al mismo paquete IPv6. sin usar tunneling. Múltiples niveles de protocolos de seguridad son implementados a través de tunneling. Con el modo de transporte, de la misma forma AH Y ESP son usados, AH debe aparecer como el primer encabezado después de IPv6, seguido por ESP. Con esta secuencia, la autenticación es asi aplicada a la salida cifrada de ESP. Con el modo de túnel órdenes diferentes de AH y ESP son posibles, dependiendo de los requerimientos de seguridad.

3.3 La Cabecera de Autentificación (AH)

Se utiliza para proporcionar integridad sin conexión y autenticación del origen de datos para datagramas IP y para proporcionar protección contra reenvíos. Este último servicio es opcional y puede seleccionarse una vez que se ha establecido la Asociación de Seguridad (SA). La AH está definida en la RFC 2402[23], proporciona autentificación a las partes de la cabecera IP que se les pueda brindar este servicio, así como también a los datos de los protocolos de las capas superiores. Sin embargo, algunos campos de la cabecera IP pueden cambiar durante el transporte, y el valor de estos campos, cuando el paquete llega al receptor, puede que no sea previsible para el emisor. Los valores de tales campos no pueden ser protegidos por AH.

Así la protección proporcionada a la cabecera IP por AH se proporciona solo a partes de la cabecera IP. AH se puede aplicar solo, o en combinación con la Carga de Seguridad Encapsulada IP (ESP), o a través de la modalidad anidada usando el modo túnel Los servicios de seguridad pueden ser suministrados a comunicaciones, entre:

[23] *RFC 2402, " Cabecera de Autenticación IP"*

- Un par de hosts,
- Un par de security gateway (SG), o
- Un security gateway y un host.

ESP puede ser usado para proporcionar los mismos servicios de seguridad, y también para proporcionar un servicio de confidencialidad (encriptación). La diferencia principal entre la autentificación proporcionada por ESP y la de AH es la extensión de la cobertura.

3.3.1 Formato de la cabecera de Autentificación

La cabecera del protocolo IPv6 inmediatamente antes de la cabecera de AH contendrá el valor 51[24]. A continuación se definen los campos que comprenden el formato de AH (ver figura 27). Todos los campos son obligatorios, es decir, que están siempre presentes en el formato de AH y se incluyen en el cálculo del Valor de Comprobación de Integridad (Integrity Check Value ICV[25]).

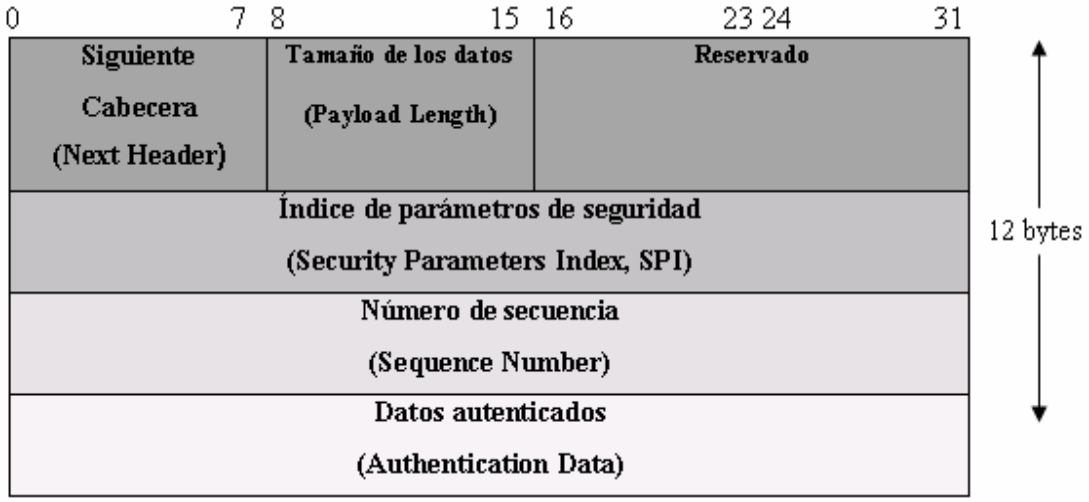


Figura 27. Formato de la cabecera de autenticación.

[24] *Ver capítulo 1 "Introducción a Ipv6", sección 5.6.3 "Los campos de la cabecera Ipv6" tabla 2 "Valores de los protocolos que pertenecen a la próxima cabecera".*
[25] *Ver capítulo 5 "Infraestructura de IPsec" sección 5.7 "Procesamiento de los paquetes de entrada y salida".*

1. Cabecera Siguiente: es un campo de 8 bits que identifica el tipo de carga siguiente después de la Cabecera de Autentificación.

2. Longitud de la Carga: posee 8 bits, especifica la longitud de AH en palabras de 32 bit (en unidades de 4 byte), menos "2". Todas las cabeceras de extensión de IPv6, según el RFC 2460[26], codifican el campo "Longitud de la Cabecera de Extensión" primero restando uno (palabra de 64-bit) a la longitud de la cabecera (medido en palabras de 64-bit). AH es una cabecera de extensión IPv6. Sin embargo, puesto que su longitud se mide en palabras de 32 bit, la "longitud de la carga" es calculada restando 2 (palabras de 32 bit). En el caso "estándar" de un valor de autentificación de 96 bits positivos divididos en 3 palabras de 32 bits de tamaño fijo, este campo tendrá una longitud de "4".

3. Reservado: Este campo de 16 bits esta reservado para uso futuro.

4. Índice de Parámetros de Seguridad (SPI): es un valor arbitrario de 32 bits que, conjuntamente con la dirección de destino IP y el protocolo de seguridad (AH), identifican unívocamente a la Asociación de Seguridad para este datagrama. El conjunto de valores de SPI en el rango de 1 a 255 se encuentran reservados para uso futuro.

5. Número de Secuencia: campo de 32 bits, es obligatorio y debe estar siempre presente incluso si el receptor elige no habilitar el servicio de antireplay para una SA específica. El procesamiento del campo Número de Secuencia esta a criterio del receptor, es decir, el emisor debe transmitir siempre este campo, pero el receptor no necesita actuar sobre él

6 Datos de Autenticación: este campo es de longitud variable y contiene el Valor de Comprobación de Integridad (ICV) para este paquete. Este campo debe contener un múltiplo entero de 32 bits de longitud. Puede incluir relleno explícito que se incluye para asegurarse de que la longitud de la cabecera de AH sea múltiplo entero de 32 bits (en IPv4) o de 64 bits (en IPv6).

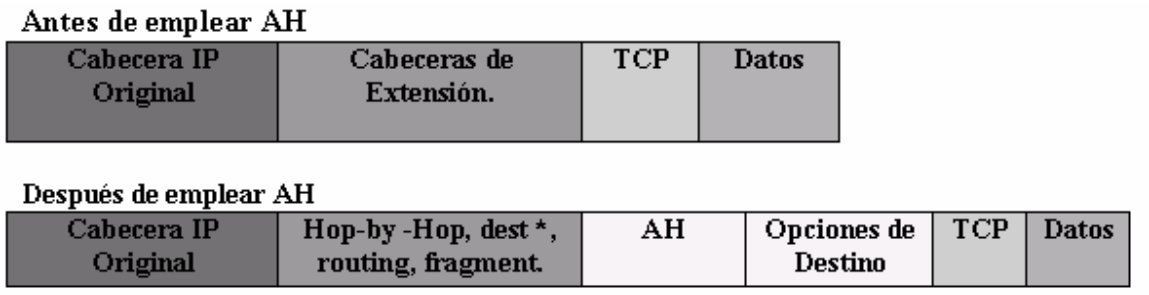
[26] *RFC 2460, "Especificaciones del Protocolo Internet Versión 6 (IPv6)"*

3.3.2 Localización de la Cabecera de Autentificación.

Al igual que ESP, el HA se puede emplear en modo trasporte o en modo túnel.

a) Modo Transporte:

El modo trasporte es aplicable solamente a implementaciones host y proporciona protección para los protocolos de capa superiores, además de los campos seleccionados de la cabecera IP. En modo trasporte, AH se inserta después de la cabecera IP y antes del protocolo de capa superior (ver figura 28), por ejemplo, TCP, UDP, ICMP, etc. o antes de cualquier otra cabecera IPsec que ya se haya incluido. En el contexto IPv6, el AH se visualiza como carga extremo a extremo (end-to-end), y debe aparecer después de las cabeceras de extensión (Véase sección 2.6.4): salto-por-salto (hop-by-hop), de encaminamiento (routing), y de fragmentación. La figura siguiente ilustra AH en modo trasporte colocado en un paquete típico de IPv6.



*si están presentes, pueden estar antes que AH, después de AH, o en ambos

Figura 28 Localización de la cabecera AH en modo transporte.

b) Modo túnel:

Puede ser empleada en host o securitys gateway. Cuando AH se implementa en una security gateway (protege tráfico en tránsito), el modo túnel debe ser utilizado. En modo túnel, la cabecera IP "interna" lleva la última dirección de origen y de destino, mientras que la cabecera IP "externa" puede contener distintas direcciones IP, por ejemplo, direcciones de securitys gateway. En modo túnel, AH protege el paquete IP interno completamente, incluyendo la cabecera IP interna entera. La posición de AH en

modo túnel, concerniente a la cabecera IP exterior, es igual que para AH en modo trasporte. La figura 29, ilustra AH en modo túnel colocado en un paquete de IPv6.

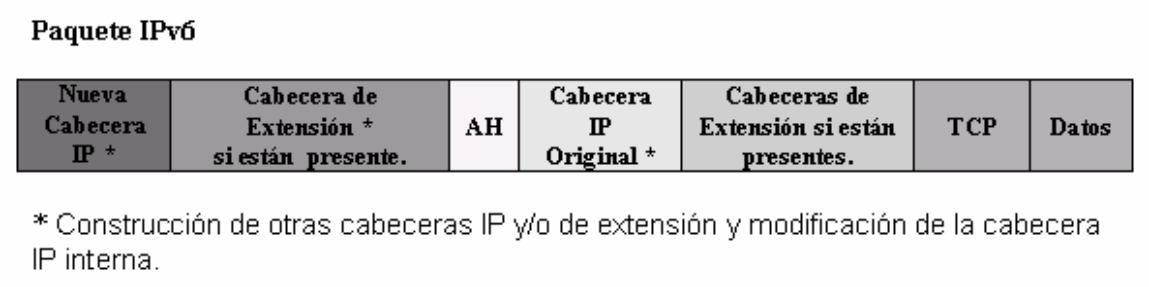


Figura 29 Localización de la cabecera AH en modo túnel.

3.4 La cabecera de Carga de Seguridad Encapsulada (ESP).

Está diseñada para proporcionar un conjunto de servicios de seguridad en IPv4 y en IPv6. ESP está definido en la RFC 2406[27] y puede ser aplicado solo, o en conjunto con la Cabecera de Autentificación (AH). Los servicios de seguridad pueden ser suministrados a comunicaciones, entre un par de hosts, o entre un par de security gateway (SG), o entre security gateway y un host.

La cabecera ESP se inserta antes que la cabecera IP y después que la cabecera de protocolo de capa superior (en modo transporte) o después de una cabecera IP encapsulada (en modo túnel). ESP es usado para proporcionar confidencialidad, autentificación del origen de los datos, integridad sin conexión, un servicio de anti-replay y confidencialidad limitada del flujo de tráfico.

El conjunto de servicios proporcionados depende de las opciones seleccionadas al momento del establecimiento de la Asociación de Seguridad (SA) y de dónde esté localizada la implementación. La confidencialidad puede ser seleccionada independientemente del resto de los servicios. No obstante el uso de la confidencialidad sin integridad/autentificación (en ESP o en AH) puede subordinar tráfico hacia ciertos tipos de ataques activos que podrían socavar el servicio de confidencialidad. Las

[27] *RFC 2406, "Encriptación de datos en IP (ESP)"*

opciones requeridas actualmente para el manejo de claves tanto para AH como para ESP son el modo manual y en el modo automatizado por medio de IKE[28] (véase el Capítulo 6).

3.4.1 Formato del Paquete de la Carga de Seguridad Encapsulada

La cabecera del protocolo inmediatamente antes de la cabecera de ESP contendrá el valor 50 en el campo Siguiente Cabecera (Ver tabla 2, pág. 28). El formato del paquete de la carga de seguridad encapsulada está compuesto por los siguientes campos (Ver figura 30):

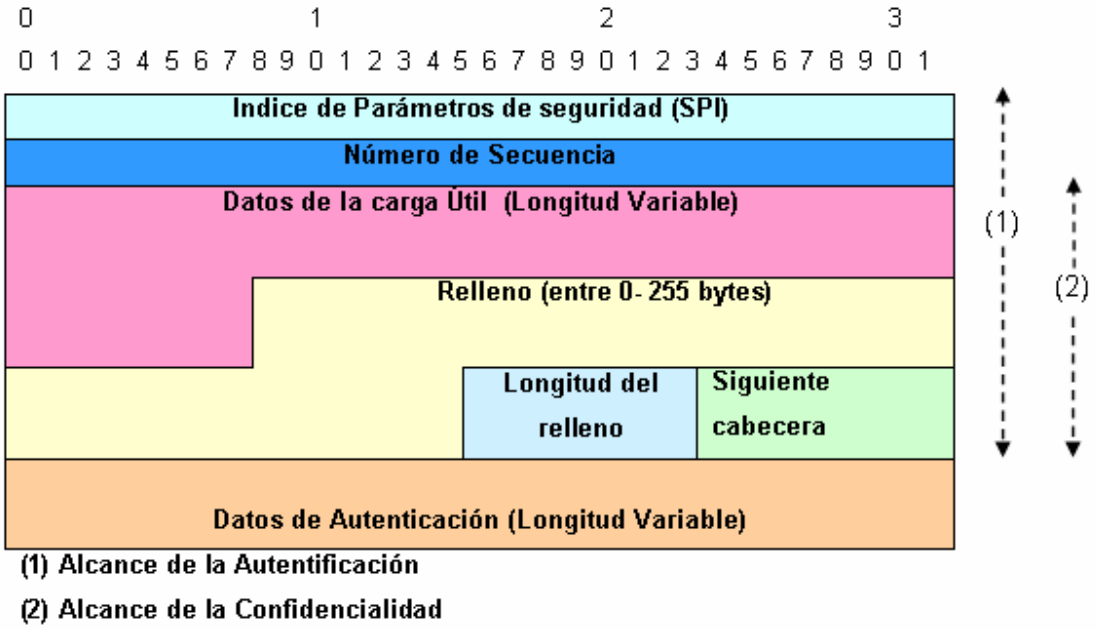


Figura 30 Cabecera ESP

1. Índice de Parámetros de Seguridad (SPI): es un valor arbitrario de 32 bits que, conjuntamente con la dirección de destino IP y el protocolo de seguridad (ESP), identifican unívocamente a la Asociación de Seguridad para este datagrama. El conjunto de valores de SPI en el rango de 1 a 255 se encuentran reservados.

[28] IKE es un protocolo de intercambio de claves por Internet.

Este es seleccionado por el sistema de destino sobre el establecimiento de una SA. El campo SPI es obligatorio. El valor de SPI cero (0) esta reservado para usarse localmente, las implementaciones no deben trasmitir este valor por la red. Por ejemplo, una implementación de administración de clave puede utilizar el valor cero de SPI para denotar que "No Existe Asociación de Seguridad" durante el período en el cual la implementación IPsec ha solicitado a la entidad administradora de claves que se establezca una nueva SA, pero la SA todavía no se ha establecido. La necesidad del SPI se hace evidente cuando tenemos más de una comunicación con la misma dirección IP de destino y protocolo de seguridad (AH o ESP).

2. Número de Secuencia: campo de 32 bits sin signo que contiene un valor creciente y único del contador (del número de secuencia). Es obligatorio y debe estar siempre presente incluso si el receptor elige no habilitar el servicio de antireplay para una SA específica. El procesamiento del campo Número de Secuencia esta a criterio del receptor, es decir, el emisor debe transmitir siempre este campo, pero el receptor no necesita actuar sobre él. El contador del emisor y del receptor se inicializan a cero (0) cuando se establece una SA.

3 Datos de la Carga Útil: es un campo de longitud variable que contiene los datos descritos por el campo Siguiente Cabecera. El campo Datos de la Carga Útil es obligatorio y su longitud es un número de bytes enteros. Si el algoritmo usado para encriptar a la carga útil requiere datos de sincronización criptográficos, por ejemplo, de un Vector de Inicialización (IV)[29] , estos datos se pueden llevar explícitamente en el campo Carga Útil. Cualquier algoritmo de encriptación que requiera tales datos explícitos, debe de enviar un paquete previamente de sincronización indicando: la longitud, la estructura para tales datos, y la localización de estos datos de sincronización criptográficos como parte de la especificación del RFC 2406 [30] del algoritmo que se utiliza con ESP.

4 Relleno (para la Encriptación): varios factores requieren o motivan el uso del campo Relleno, algunos de ellos son:

---

[29] *Un **vector de inicialización** (conocido por sus siglas en inglés **IV**) es un bloque de bits que es requerido para permitir un cifrado en flujo, en uno de los modos de cifrado, con un resultado independiente de otros cifrados producidos por la misma clave.*

[30] *RFC 2406, "Encriptación de datos en IP (ESP)"*

- Si se emplea un algoritmo de encriptación que requiere que el texto plano sea un múltiplo de un cierto número de bytes, por ejemplo, el tamaño de bloque de un bloque cifrado, el campo Relleno es usado para rellenar el texto plano (el cual consta de los Datos de la Carga Útil, y los campos Longitud del Relleno y Siguiente Cabecera, así como también del Relleno) para el tamaño requerido por el algoritmo.

- El relleno también puede ser requerido, independientemente de los requisitos del algoritmo de encriptación, para asegurarse de que el texto cifrado resultante termine en un límite de 4 bytes. Específicamente, los campos Longitud del Relleno y Siguiente Cabecera deben estar alineados correctamente dentro de una palabra de 4 bytes, según lo ilustrado en la figura del formato del paquete ESP, para asegurarse de que el campo Datos de Autentificación esté alineado en un límite de 4 bytes.

- Más allá del relleno requerido para el algoritmo o por las razones de alineación se puede utilizar para cubrir la longitud real de la carga, en respaldo de la confidencialidad del flujo de tráfico. Sin embargo, la inclusión de tal relleno adicional tiene implicaciones adversas en el ancho de banda y su uso bebe ser emprendido con cautela.

El emisor puede agregar de 0 a 255 bytes de relleno. La inclusión del campo Relleno en un paquete ESP es opcional, pero todas las implementaciones deben de soportar la generación y el uso del relleno.

5 Longitud del Relleno: indica el número de bytes de relleno inmediatamente precedentes a este campo. El rango de valores válidos es de 0 a 255 bytes, donde un valor de cero indica que no hay bytes de Relleno presentes. El campo Longitud del Relleno es obligatorio.

6 Siguiente Cabecera: es un campo de 8 bits que identifica el tipo de datos contenidos en el campo Datos de la Carga Útil, por ejemplo, una cabecera de extensión IPv6 o un identificador de protocolo de capa superior. El valor de este campo se elige del conjunto de Números de Protocolo IP definidos por la IANA[31].

7. Datos de Autentificación: campo es de longitud variable y contiene el Valor de Comprobación de Integridad (ICV) calculado sobre el paquete ESP menos, los Datos de Autentificación. La longitud del campo es especificada por la función de autentificación seleccionada. El campo Datos de Autentificación es opcional, y se incluye solamente si el servicio de autentificación se ha seleccionado para la SA en cuestión. La especificación del algoritmo de autentificación debe especificar la longitud del ICV y las reglas de comparación y los pasos para el procesamiento de validación.

3.4.2 Localización de la Cabecera ESP

a) Modo de transporte:

El primer modo es aplicable solamente a implementaciones host y proporciona la protección para los protocolos de capa superiores, pero no a la cabecera IP. En modo transporte, ESP se inserta después de la cabecera IP y antes del protocolo de capa superior, por ejemplo, TCP, UDP, ICMP, etc. o antes que cualquier otra cabecera IPsec que se haya insertado. En el contexto de IPv4, esto se traduce en la colocación de ESP después de la cabecera IP (y de cualquiera de las opciones que contenga), pero antes del protocolo de capa superior.

En el contexto de IPv6, ESP se visualiza como carga útil extremo a extremo, y por lo tanto debe aparecer después de las cabeceras de extensión de salto-por-salto (hop-by-hop), ruteo (routing) y de fragmentación. La figura 31 ilustra a ESP en modo transporte ubicado en un paquete típico de IPv6.

[31] ***Internet Assigned Numbers Authority****, es la Agencia de Asignación de Números de Internet. Era el antiguo registro central de los protocolos Internet, como puertos, números de protocolo y empresa, opciones y códigos. Fue sustituido en 1998 por ICANN.*

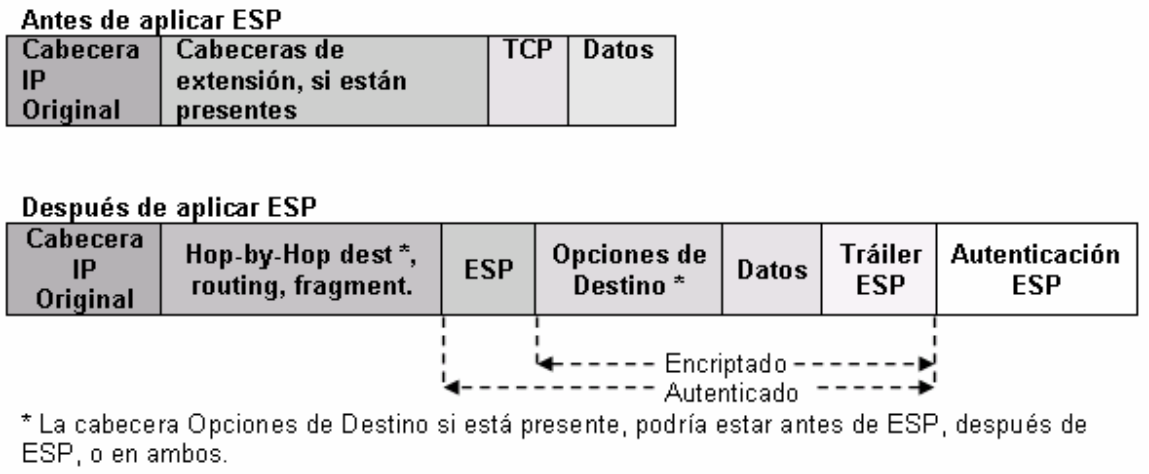


Figura 31 Localización de la cabecera ESP en modo transporte.

b) Modo túnel

Puede ser empleado en hosts o security gateways. Cuando se implementa ESP en una security gateway (para proteger el tráfico en tránsito del suscriptor), se debe utilizar el modo túnel. En modo túnel, la cabecera IP "interna" lleva las últimas direcciones de origen y de destino, mientras que una cabecera IP "externa" puede contener direcciones IP distintas, por ejemplo, las direcciones de las security gateways. En modo túnel, ESP protege a todo el paquete IP interno, incluyendo toda la cabecera IP interna. La posición de ESP en modo túnel, concerniente a la cabecera externa IP, es igual que para ESP en modo transporte. La figura 32, ilustra ESP en modo túnel ubicado en un paquete típico de IPv4 y de IPv6.

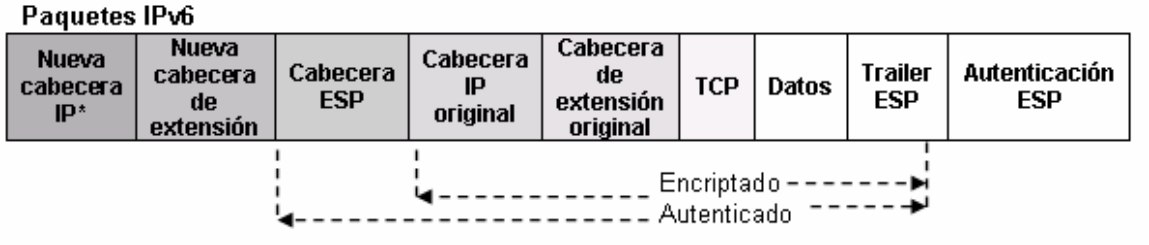


Figura 32. Localización de la cabecera ESP en modo túnel.

Capítulo

4

Redes Privadas Virtuales (VPN).

Temas:

- Redes Privadas Virtuales (VPN).
- Túneles.
- Componentes de la VPN.
- Tipos de VPN.
- Modelos de implementación VPN
- Implementación de las Redes Privadas Virtuales.
- Trabajando con IPv6.

4. Redes Privadas Virtuales (VPN)

En los últimos años se ha dado mucha importancia a la seguridad de la implementación de las redes, el uso de encriptación es común y las empresas buscan soluciones más eficaces y económicas posibles ante la inseguridad de Internet. Para dar solución a estas demandas surgieron las VPN (*Redes Virtuales Privadas)*.

La VPN es una red que ofrece conexiones seguras a través de Internet. Se trata de una extensión de red privada que utiliza enlaces a través de redes públicas o compartidas IP. Estas redes permiten conectar empleados móviles, oficinas y delegaciones separadas geográficamente, así como a socios y clientes de una forma muy segura (Ver figura 33). Las empresas obtienen de esta forma reducción de gastos, aumentan su seguridad, incrementan la calidad de servicio (QoS), habilitan las aplicaciones de voz y video, y adquieren mayor facilidad para compartir recursos entre sus delegaciones.

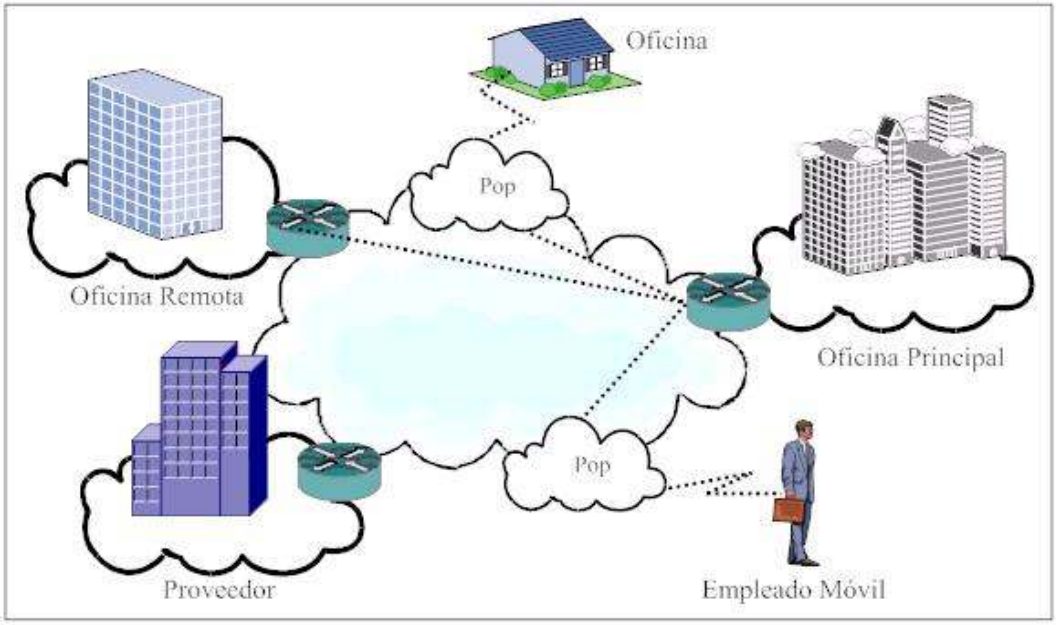


Figura 33 Diseño de una VPN, la red conecta a muchos usuarios de diferentes lugares, enlaza a la empresa principal, sus respectivas oficinas, proveedores y empleados móviles.

También permiten enviar datos entre dos extremos, simulando un enlace punto a punto mediante la encapsulación de los mismos, brinda una cabecera que contiene datos de enrutamiento. Para que el enlace se mantenga es necesario recurrir a mecanismos de seguridad como por ejemplo el cifrado de información[32].

4.1 Túneles

Un túnel entre dos sedes consiste en el método para mantener una intranet mediante el uso de una infraestructura de red pública para la interconexión y el envío de datos dentro de la propia red privada (Ver figura 34). Tunneling es el proceso de encapsular un paquete IP dentro de otro, se utiliza para crear redes privadas virtuales; la funcionalidad de los túneles es enlazar segmentos de redes que trabajan con IPv6 sobre una red que opera y funciona con el protocolo IPv4. Los túneles se pueden clasificar en dos tipos: estáticos y dinámicos.

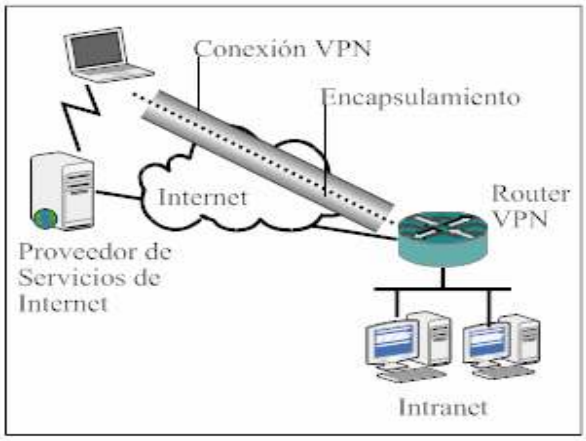


Figura 34. Construcción de un túnel

El proceso de encapsulación se ilustra en la figura 35; donde se visualiza el resultado del datagrama del protocolo IPv4, dentro de éste se encuentra el encabezado IPv6, y toda la información de las la capas superiores, por ejemplo los encabezados TCP, datos de aplicación, etc.

[32] *Ver capitulo 4 "Mecanismos en la seguridad en Ipv6"*

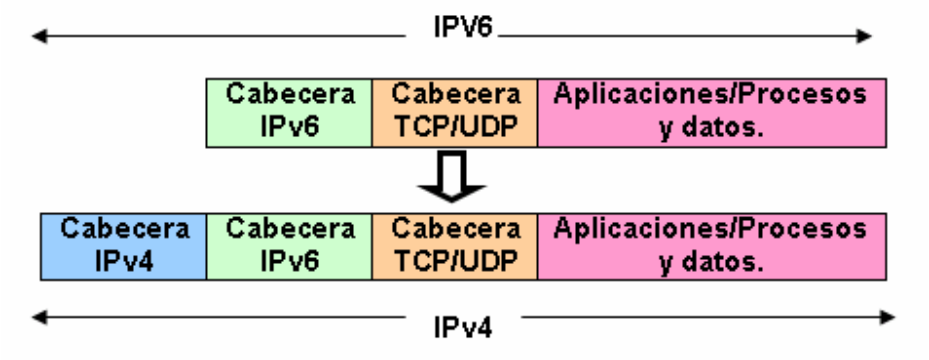


Figura 35. Proceso de encapsulación.

Cuando el destino recibe los datagramas correctos, entonces realiza el proceso inverso de encapsulación, el cual se le llama desencapsulación, tal como se muestra en la figura 36, se observa que a partir del encabezado IPv4, se extrae el encabezado IPv6, el valor del campo del protocolo obtendría un valor de 41, para identificar que se trata del protocolo IPv6[33].

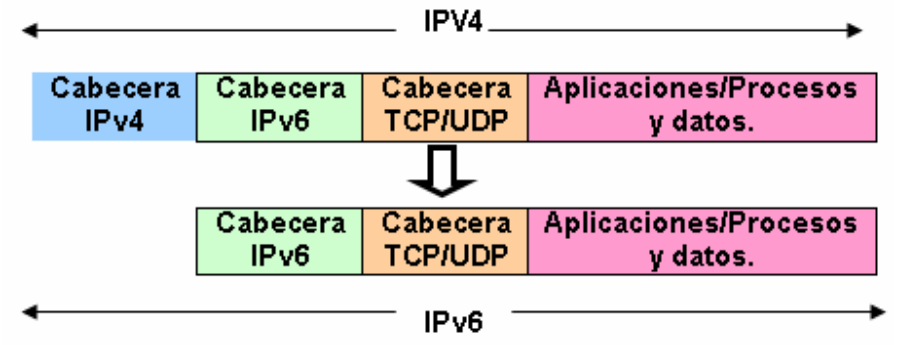


Figura 36 . Proceso de desencapsulación.

4.1.1 Tipos de túneles:

a) Estáticos:

Se emplean cuando un host que trabaja con una dirección IPv4 y desea tener acceso a la red de IPv6 existente. Para ese caso, se implementa un túnel con un enrutador a través de IPv4 que tenga acceso a IPv6 como a IPv4. El 6bone es un banco de pruebas del protocolo IPv6 creado por la IETF.

[33] *Ver capítulo 1 "Introducción a Ipv6", sección "Los campos en la cabecera de Ipv6", tabla2.*

En la actualidad el 6bone es un proyecto de colaboración mutua de alcance mundial. En sus comienzos era una red virtual creada sobre la Internet existente, empleando túneles; pero con el tiempo muchas partes del 6bone han sido migradas a IPv6 nativo. Cuando el 6bone funcionaba como una VPN, los túneles que se utilizaban eran en su mayoría de tipo estáticos.

b) 6to4:

Este mecanismo se puede aplicar para comunicar redes IPv6 aisladas por medio de la red IPv4. El enrutador extremo de la red IPv6 crea un túnel sobre IPv4 para alcanzar la otra red IPv6. Los extremos del túnel son identificados por el prefijo del sitio IPv6.

c) 6over4:

Si no se tiene una red homogénea en el aspecto de que todos los nodos puedan comunicarse entre si con la misma versión del protocolo IP, con este método se puede comunicar nodos IPv6 aislados dentro del sitio con el resto de nodos IPv4. Esta técnica también se emplea en casos en los cuales el enrutador IPv6 no tenga acceso o permiso para transmitir paquetes IPv6 sobre enlaces.

4.2 Componentes de la VPN.

Una VPN se construye sobre una infraestructura con funcionalidades de red y de seguridad equivalentes a las que se obtienen con una red privada real. El objetivo de la VPN es el soporte de aplicaciones Intranet y extranet, integrando aplicaciones multimedia de voz, datos y video sobre infraestructuras de comunicaciones eficaces.

Las primeras redes de computadoras fueron implementadas con dos tecnologías fundamentales: líneas dedicadas; para una conectividad permanente y conexiones conmutadas; para requerimientos de conectividad ocasional. Estas redes iniciales brindaban seguridad a los usuarios pero no ofrecían una buena relación entre el costo y beneficio por dos razones fundamentales:

- El promedio de tráfico entre dos sitios de una red varía debido a muchos factores, entre ellos, el momento del día, de la semana, del mes, etc.
- Los usuarios finales requieren de respuestas rápidas, lo cual implica que se debe de tener altos anchos de banda entre los sitios de red. Pero el ancho de banda de una línea dedicada sólo es usado una parte del tiempo, cuando el usuario se encuentra activo.

Estas dos razones iniciales llevaron a la industria del transporte de datos y a los proveedores de servicio a desarrollar e implementar esquemas de redes de conmutación de paquetes con principios de multiplexación que brindan a los clientes servicios equivalentes a líneas dedicadas. Las primeras soluciones de este tipo de conectividad fue implementada basada en X.25, luego Frame Relay y más tarde ATM. La figura 37 muestra una red VPN típica construida con tecnologías Frame Relay:

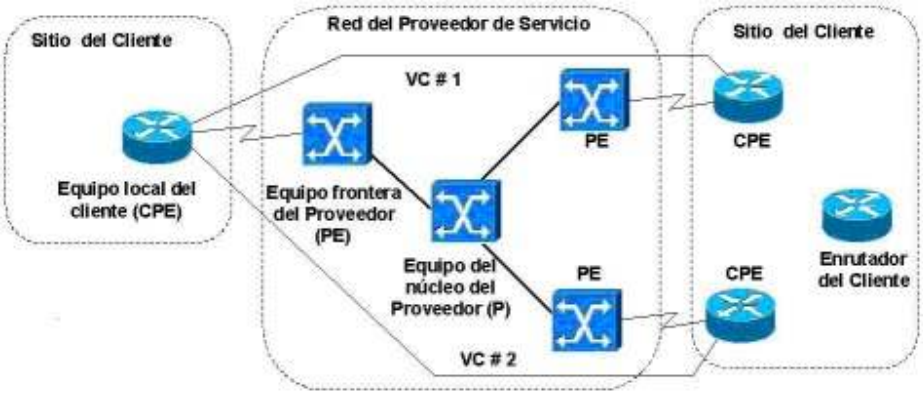


Figura 37. Ejemplo de VPN sobre Frame Relay.

Se puede observar que las soluciones VPN tienen como componentes:

SP: proveedor de servicio, es la organización propietaria de la infraestructura (el equipamiento y el medio de transmisión) con la que brinda emulación de líneas dedicadas a los clientes.

CPE: (*Customer Premises Equipment*), es la conexión del cliente a la red del SP a través del equipo local del cliente. El CPE es conectado a través de un medio de transmisión,

equipo del SP el que puede ser X.25, *Frame Relay*, ATM o un enrutador IP; el dispositivo CPE es a veces llamado también frontera del cliente (**CE**, *Customer Edge*).
PE: dispositivo de frontera del SP es llamado, *Provider Edge*. Generalmente tiene equipos de núcleo en la red, los que son llamados **P**, *Provider*.

Conexión VPN: porción de la conexión entre dos usuarios, donde se encriptan los datos. Una conexión de acceso remoto sucede cuando un ordenador personal se conecta a la red privada para poder obtener los servicios del servidor o para acceder a la red completa. Una conexión enrutador-enrutador conecta dos porciones de la red privada. En ambos casos el servidor pide autenticación del cliente. La autenticación puede ser mutua.

Túnel. El fragmento de conexión que está dentro de la red VPN de la organización, en el cual se encapsulan los datos para que al salir a la red IP se transporten seguros hasta su destino.

Protocolos de Túnel: Los protocolos empleados en estas redes son: PPTP (túneles Punto- Punto), IPSec (Protocolo de Internet de Seguridad), L2TP (Protocolo de túneles de Capa de enlace de datos del modelo OSI [34].

Otros conceptos importantes en el desarrollo de las redes privadas virtuales son:

VPN IPv4: es una VPN que conecta y hablita a clientes que trabajan con IP4.

VPN IPv6: es una VPN que conecta y hablita a clientes que trabajan con IP6.

Doble pila VPN: es una VPN que conecta a VPN basada en IPv6 y VPN que trabajan con el protocolo IPv4.

VPN hibrida: es una VPN que conecta y habilita solamente IPv4 o IPv6, pero no una mezcla de ambos protocolos.

[34] *Ver anexo I*

4.3 Tipos de VPN.

A continuación, se expone una clasificación según: tipos de IP VPN dados en la RFC 2764[35], alcance de las VPN para las organizaciones, modelos implementados superpuestos, y par a par.

4.3.1 Tipos de IP VPN según RFC 2764.

a. VLL (Virtual Leased Lines): Líneas Dedicadas Virtuales brindan enlaces punto a punto orientados a conexión entre los sitios de los clientes. El cliente percibe cada VLL como un enlace (físico) privado dedicado.

b. VPLS (Virtual Private LAN Segments): Segmentos LAN Privados Virtuales brinda una imitación de LAN entre sitios VPLS, como con la VLL un VPLS requiere del uso de túneles IP que sean transparentes a los protocolos transportados por las LAN simuladas.

c. VPRN (Virtual Private Routed Networks): Redes de Enrutadores Privados Virtuales simulan redes dedicadas de enrutadores IP entre los sitios de los clientes, aunque una VPRN transporte tráfico IP.

d. VPDN (Virtual Private Dial Networks): Redes Conmutadas Privadas Virtuales, les permite a los clientes que el SP le gestione los accesos conmutados a su red. En lugar de cada cliente configurar sus propios servidores de acceso y usando secciones PPP (Point to Point Protocol) entre un local central y los usuarios remotos, el SP brinda uno o muchos servidores de acceso compartidos.

[35] *RFC 2764 Esquema para IP basado en Redes Privadas Virtuales.*

4.3.2 Según el alcance de la VPN para la organización.

Las categorías siguientes no son excluyentes, es decir, una red puede estar conformada por una combinación de ellas, incluso por la unión de las tres categorías; VPN intranet, VPN extranet y accesos remotos.

a. VPN intranet, entre departamentos de una misma organización: se utilizan para interconectar departamentos o dependencias de una misma organización son generalmente redes con un alto nivel de aislamiento y seguridad, además requieren de garantías de calidad deservicio para aplicaciones críticas; utilizan Internet para este tipo de VPN. (Ver figura 38).

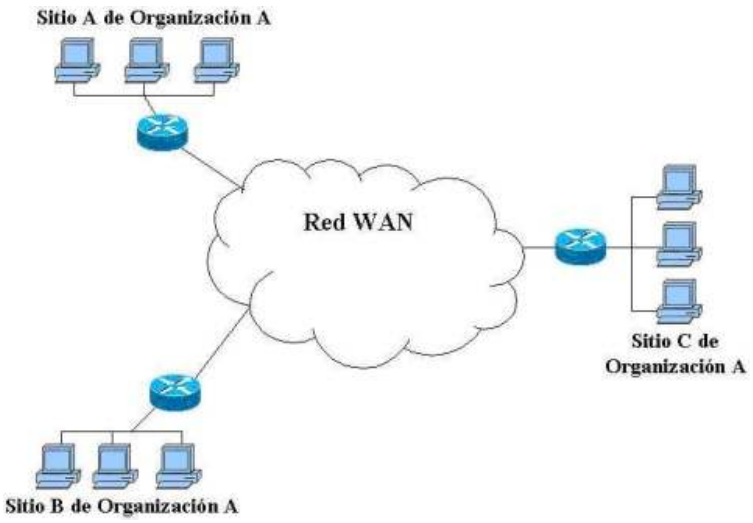


Figura 38. VPN como intranet.

b. VPN extranet, entre una organización, sus socios, clientes y suministradores: Interconectan sitios principales de diferentes organizaciones, usualmente dedicando dispositivos de seguridad como firewall o de encriptación, similar a la configuración mostrada en la figura 39.

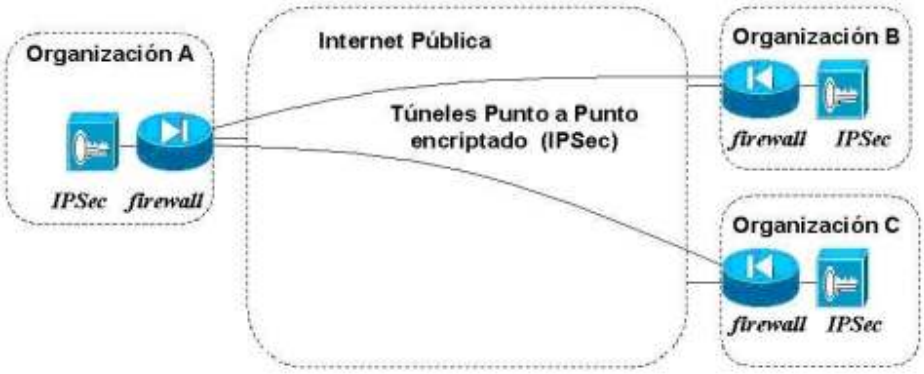


Figura 39 Configuración típica de una VPN extranet utilizando Internet.

c. VPN con accesos remotos, entre la organización y empleados móviles o remotos: Presentan características similares a las VPDN de la RFC 2764 descritas anteriormente y utilizan protocolos como L2F (Layer 2 Forwarding) o L2TP (Layer 2 Tunneling Protocol (Ver figura 40).

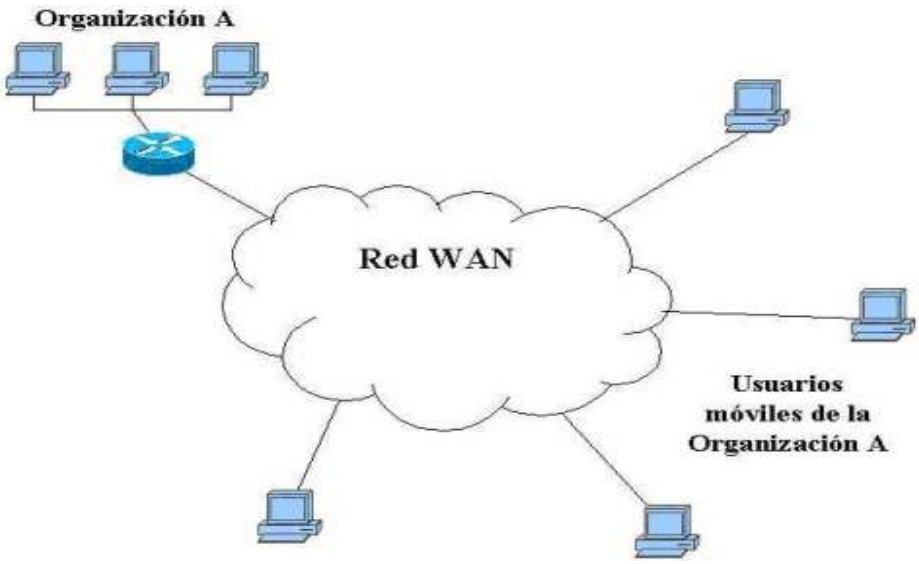


Figura 40. VPN con accesos remotos.

4.4 Modelos de implementación VPN

Modelo superpuesto (*overlay*), donde el SP simula líneas dedicadas para el cliente y el modelo par a par (*peer to peer*) donde el SP y el usuario intercambian información de enrutamiento, el proveedor transporta los datos entre los sitios del usuario por un trayecto óptimo.

a. Modelo superpuesto: el SP brinda al cliente una configuración que simula líneas dedicadas llamadas circuitos virtuales VC los que pueden estar disponibles constantemente (PVC, Permanente Virtual Circuit) o establecidos bajo demanda (SVC, Switched Virtual Circuit). (Ver figura 41).

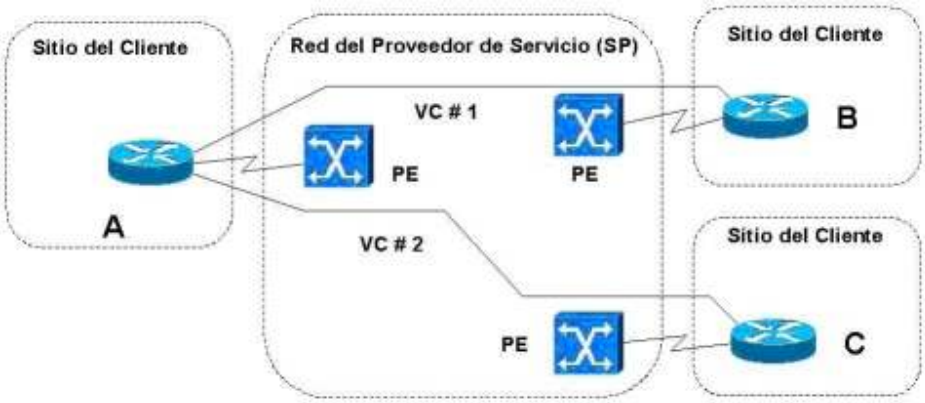


Figura 41 Ejemplo de topología de red VPN superpuesta.

Como se observa en la Figura 44 el cliente establece comunicación entre sus enrutadores sobre los VCs suministrados por el SP. La información de los protocolos de enrutamiento siempre es intercambiada entre los dispositivos del cliente por lo que el SP desconoce la topología interna de la red del cliente.

b. Modelo par a par el PE es un enrutador que intercambia directamente información de enrutamiento con el CPE. La figura 42 muestra un ejemplo de VPN par a par.

Figura 42. Ejemplo de VPN par a par.

4.5 Implementación de las Redes Privadas Virtuales

Para su implementación se debe de analizar el sistema que en la actualidad predomina en las comunicaciones, y éste es el modelo OSI (Open Systems Interconnection). Dentro de este modelo de interconexión, como se puede observar en la figura 43, sigue un flujo continuo de datos.

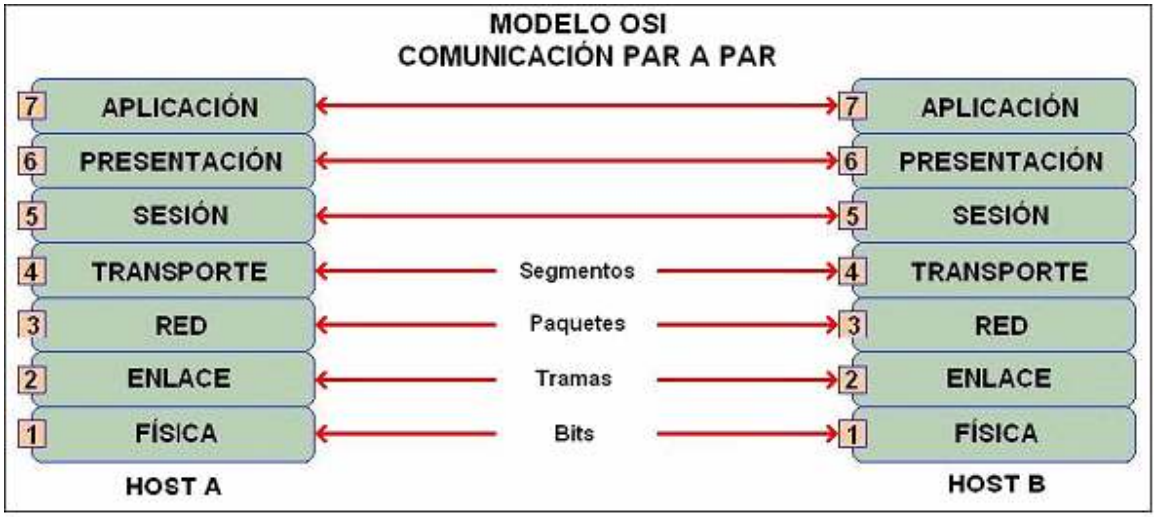


Figura 43: Modelo OSI

El proceso de encriptación y desencriptación de la información se puede realizar en cualquier punto del flujo de la información, sólo con la restricción de realizar los procesos referidos en las mismas capas equivalentes. Por consiguiente, atendiendo al modelo OSI, se puede observar dos áreas importantes: Hardware y Software. El término Hardware se refiere al sistema de interconexión física de los dos equipos (capa física), mientras que el término Software se aplica a las otras capas del modelo OSI. Dado que la encriptación y desencriptación se puede realizar en los puntos que se deseen, siempre y cuando sean capas equivalentes, se puede seleccionar estos dos puntos de implementación definidos, VPN por Hardware y VPN por Software.

4.5.1 Implementación de VPN por Hardware

El proceso de encriptación y desencriptación se realiza a nivel físico en los puntos Inmediatamente anterior e inmediatamente posterior al comienzo de la línea de comunicación. Por realizarse a nivel físico, se necesita equipos que permitan realizar esta tarea de forma transparente. Por lo general los elementos utilizados son los enrutadores con VPN incorporada.

Estos dispositivos llevan incorporado un procesado y algoritmos de encriptación y desencriptación. Tienen la ventaja de que el fabricante brinda la implementación adecua y su respectiva instalación. Las VPN implementadas por hardware, presentan el inconveniente, de que el sistema de encriptación viene impuesto por el fabricante, y depende del mismo para las actualizaciones.

Dentro de esta categoría se puede incluir los enrutadores wireless con encriptación, mediante WPA[36] y WEP[37], debido a que crean un túnel entre el enrutador y la tarjeta wireless que impiden en cierta forma la lectura y modificación de la información. En este caso el medio de transporte son las ondas electromagnéticas y por tener el enrutador y la tarjeta inalámbrica, la encriptación se realiza a nivel de capa física. Las ventajas e inconvenientes que presenta este tipo de configuración son:

[36] *Wi-Fi Protected Access (Acceso Protegido Wi-Fi); es un sistema para proteger las redes inalámbricas.*
[37] ***Wired Equivalent Privacy**, es el sistema de cifrado incluido en el estándar IEEE 802.11 como protocolo para redes Wireless que permite cifrar la información que se transmite.*

Ventajas:

- La instalación y la configuración poseen menos dificultad.
- No se necesita personal especializado y su mantenimiento es mínimo.
- Un único elemento puede habilitar varias VPN ubicadas en distintos sitios.
- El sistema es independiente de las máquinas conectadas a la red.
- No se necesita de máquinas dedicadas para realizar la VPN.

Inconvenientes:

- Depende de una tecnología externa y cerrada.
- Existe una dependencia con el fabricante para poder realizar cambios.
- Los sistemas de encriptación suelen ser cerrados y el fabricante suele utilizar un único tipo.
- La seguridad sólo se implementa desde los dos extremos de la VPN, siendo inseguro el camino que recorre la información desde el ordenador hasta el dispositivo VPN.
- En la mayoría de las ocasiones los elementos hardware de los extremos que componen la red privada virtual, deben ser iguales o por lo menos del mismo fabricante.
- Sólo se utilizan para realizar conexiones VPN dentro de la misma red (intranet) o sólo fuera de la red, pero no pueden realizar simultáneamente las dos opciones.

4.5.2 Implementación de VPN por Software

Cada día crece la demanda de la utilización de Redes Privadas Virtuales por software. La razón de ésta necesidad es que los medianos y los pequeños usuarios implementan sistemas de seguridad en el acceso a sus máquinas. Además los sistemas que ocupan tienden a crecer, entonces es económico utilizar VPN por software que por hardware. Las ventajas y desventajas que pueden presentar este tipo de redes son:

Ventajas:

- Existe una gran variedad de Redes Privadas Virtuales desarrolladas por software.
- El número de usuarios de este tipo de red es mucho mayor que el número de usuarios de VPN realizadas por hardware.
- Ofrecen cobertura tanto a redes internas (intranet) como redes externas.
- La seguridad puede cubrir de máquina a máquina, donde se encuentren colocados los extremos de la VPN.

Inconvenientes:

- El sistema de claves y certificados están en máquinas potencialmente inseguras, que pueden ser atacadas.

4.5.2.1 Tipos de VPN por Software

Existe una gran variedad de tipos de Redes Privadas Virtuales por Software. Entre ellas se pueden mencionar: IPSec[38], PPTP [39], L2TP[40], VPN SSL/TLS, Open VPN.

a. IPSec

Es la abreviatura de Internet Protocol Security, inicialmente se desarrolló para usarse con el estándar IPv6 y posteriormente se adaptó a IPv4. Es una extensión al protocolo IP. Añade los servicios de autenticación y cifrado. IPSec actúa dentro del modelo OSI en la capa 3 (capa de red). No está ligado a ningún algoritmo de encriptación o autenticación, tecnología de claves o algoritmos de seguridad especifico. De hecho es un estándar que permite que cualquier algoritmo nuevo se pueda introducir. Por sus características es considerado como el protocolo estándar para la construcción de redes privadas virtuales La especificación del protocolo se encuentra en la RFC 2401. IPSec cuenta con dos protocolos diferentes, estos son:

[38] *Es una extensión al protocolo IP que añade cifrado para permitir servicios de autenticación , asegura las comunicaciones a través de dicho protocolo.*
[39] *Desarrollado por Microsoft, U.S. Robotics, Ascend Communications, 3Com/Primary Access, ECI Telematics conocidas colectivamente como PPTP Forum, para implementar redes privadas virtuales o VPN..*
[40] *Fué diseñado por un grupo de trabajo de, creado para corregir las deficiencias entre PPTP y l2F.*

- Cabecera de Autenticación (Authentication Header, AH). Se trata de una nueva cabecera que se obtiene el protocolo básico IP y que se añade a los resúmenes criptográficos de los datos e información de identificación.
- Encapsulado de Seguridad (Encapsulating Security Payload, ESP). Permite reescribir los datos en modo cifrado. No considera los campos de la cabecera IP por lo que sólo garantiza la integridad de los datos.

Ambos protocolos controlan el acceso y distribuyen las claves criptográficas. No pueden ser aplicados los dos a la vez. Lo que sí se permite es aplicarlos uno después de otro, es decir, a un datagrama IP aplicarle un protocolo y al paquete resultante aplicarle otro. Si se hace esto el orden de aplicación es: ESP-AH Cada uno de estos protocolos pueden funcionar en dos modos distintos: modo transporte, modo túnel.

- El modo transporte es el que usa un anfitrión que genera los paquetes.
- En modo transporte, las cabeceras de seguridad se añaden antes que las cabeceras de la capa de transporte (TCP, UDP), antes de que la cabecera IP sea añadida al paquete.

b. PPTP

PPTP (Point to Point Tunneling Protocol) es un protocolo desarrollado por Microsoft, U.S. Robotics, Ascend Communications, 3Com/Primary Access, ECI Telematics conocidas colectivamente como PPTP Forum, para implementar VPN. La seguridad de PPTP ha sido completamente quebrantada, el fallo de PPTP es causado por errores de diseño en la criptografía en los protocolos handshake o apretón de manos LEAP[41] de Cisco y MSCHAP[42]-v2 de Microsoft y por las limitaciones de la longitud de la clave en MPPE. La actualización de PPTP para las plataformas Microsoft viene por parte de L2TP o IPsec. Su adopción es lenta porque PPTP es fácil de configurar, mientras L2TP requiere certificados de clave pública, e IPsec es complejo y poco soportado por plataformas antiguas como Windows 98 y Windows Me.

[41] *Lightweight Extensible Authentication Protocol. patentado por Cisco basado en nombre de usuario y contraseña que se envía sin protección.*
[42] *Protocolo de Autenticación por Desafío Mutuo, Protocolo de autenticación utilizado por el acceso remoto de Microsoft y conexiones de red y de acceso telefónico.*

c. L2TP

L2TP (Layer 2 Tunneling Protocol), fue diseñado por un grupo de trabajo de IETF como el heredero aparente de los protocolos PPTP y L2F[43], creado para corregir las deficiencias de estos protocolos y establecerse como un estándar. El transporte de L2TP está definido para una gran variedad de tipos de paquete, incluyendo X.25, Frame Relay y ATM [44].

A pesar de que L2TP ofrece un acceso económico, soporte multiprotocolo y acceso a redes de área local remotas, no presenta características criptográficas especialmente robustas. Sólo se realiza la operación de autenticación entre los puntos finales del túnel, pero no para cada uno de los paquetes que viajan por él. Sin comprobación de la integridad de cada paquete, sería posible realizar un ataque de negación del servicio por medio de mensajes falsos de control que den por acabado el túnel L2TP o la conexión PPP subyacente.

L2TP no cifra en principio el tráfico de datos de usuario, lo cual puede generar problemas cuando sea importante mantener la confidencialidad de los datos. A pesar de que la información contenida en los paquetes PPP puede ser cifrada, este protocolo no dispone de mecanismos para generación automática de claves, o actualización automático de claves.

A causa de estos inconvenientes las empresas toman la decisión de utilizar los propios protocolos IPSec para proteger los datos que viajan por un túnel L2TP.
L2TP es en realidad una variación de un protocolo de encapsulamiento IP. Un túnel L2TP se crea encapsulando una trama L2TP en un paquete UDP, el cual es encapsulado a su vez en un paquete IP, cuyas direcciones de origen y destino definen los extremos del túnel. Siendo el protocolo de encapsulamiento más externo IP, los protocolos IPSec pueden ser utilizados sobre este paquete, protegiendo así la información que se transporta por el túnel.

[43] *Layer 2 Forwarding*, diseñado por Cisco para establecer túneles de tráfico desde usuarios remotos hasta sus sedes corporativas
[44] *Modo de Transferencia Asíncrona o Asynchronous Transfer Mode (ATM) es una tecnología de telecomunicación*

d. VPN SSL/TLS

SSL/TLS Secure Sockets Layer/Transport Layer Security; SSL proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía. Por lo general sólo el servidor es autenticado, mientras que el cliente se mantiene sin autenticar; la autenticación mutua requiere de un despliegue de infraestructura de claves públicas (PKI) para los clientes. SSL implica una serie de fases básicas: Negociar entre las partes el algoritmo que se usará en la comunicación. Intercambio de claves públicas y autenticación basada en certificados digitales, encriptación del tráfico basado en cifrado simétrico[45].

4.6 Trabajando con IPv6.

En IPv4 se utiliza la técnica de tuneling IP, donde los paquetes IP son protegidos y encapsulados dentro de otro paquete IP , el cual se utiliza para transportar el paquete original y atravesar la red publica hasta llegar a su destino final. Usualmente los puntos finales de un túnel IP no son redes que quieren intercambiar los datos, por lo general son firewalls, que protegen los hosts locales de ataques externos, así como se muestra en la figura 44. La complejidad de construir una VPN utilizando el protocolo IPv6 es menor que crearla con IPv4, además es más estándar que en IPv4; esto se debe a las cabeceras AH y ESP que se incorporan en IPv6.

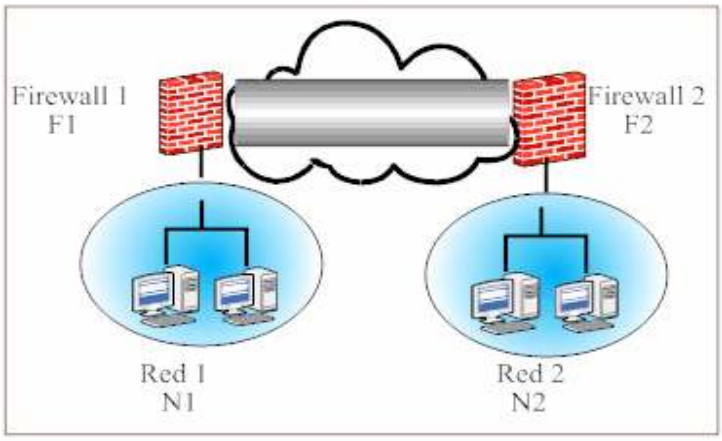


Figura 44. Ejemplo de un túnel entre dos firewalls.

[45] *Ver capítulo 2 "Seguridad en IPv6"s, sección 2.3.5 "Clave privada (simétrica)"*

En la figura 44 se presenta un canal TCP, que se encuentra entre un host H1 en la red N1, y el host H2 en la red N2. Los nodos tienen que protegerse ante las situaciones de manipulación de datos y falsificaciones existentes.

E n este caso, la cabecera AH dentro del paquete que se transporta en el canal, el firewall 1 FW1, ejecuta el paquete IP así como se muestra en la figura 45, previamente modifica una cabecera AH para poder enviar sin dificultad el paquete hacia la pareja o destino, el FW2, ejecuta el proceso de verificación de la integridad, originalidad y autenticación del paquete, con la finalidad que el destino pueda obtener los datos correctos. Ver figura 46

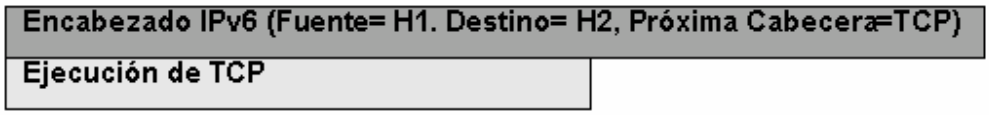


Figura 45. Paquete IPv6 enviado desde la cabecera H1 hacia el firewall F1

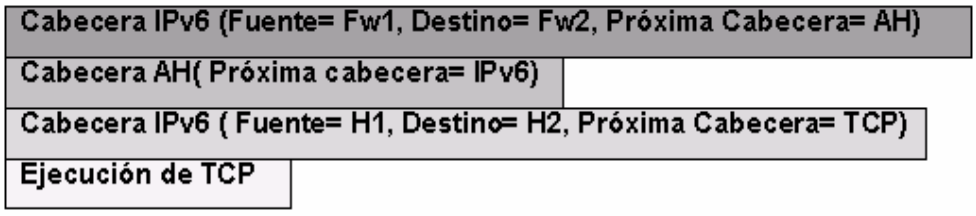


Figura 46. Paquete IPv6 enviado desde el Firewall1 Fw1 hacia el firewall 2 FW2.

Si la VPN es implementada para trabajar solamente con la cabecera AH, entonces los atacantes no pueden alterar la transmisión de los paquetes o insertar paquetes en el canal. Sin embargo, ellos pueden leer el contenido de los paquetes. Para prevenir esta situación, se utiliza y ejecuta la cabecera ESP.

Durante el transporte de los datos, los paquetes tienen el riesgo de ser eliminados por intermediarios, estos ataques no son fácil de contrastar en el nivel del protocolo IP, las

defensas apropiadas como el uso de identificadores de paquetes es usualmente utilizada en algunos niveles de las capas superiores de modelo OSI.

Si se implementan muchos firewalls, se corre el riesgo de tener problemas de compatibilidad entre los firewalls de diferentes proveedores, como consecuencia se deriva otro problema con respecto a la fragmentación, si el paquete transmitido realmente posee las máximas dimensiones permitidas por el paquete IP, entonces el encapsulamiento dentro de otro paquete IP, no es posible, la fragmentación y el reensamblamiento debe de ejecutarse en los dos puntos finales del túnel.

Capítulo

5

Infraestructura de IPSec.

Temas:

- Infraestructura de IPsec.
- Ventajas y Limitaciones.
- Componentes de IPSec.
- Cabeceras y algoritmos.
- Modos de transporte.
- El protocolo ISAKMP.
- Internet Key Exchange (IKE).
- Procesamiento de los paquetes de entrada y salida.
- Proceso de negociación y filtrado.
- Directivas de seguridad de IP.

5. Infraestructura de IPsec.

IPsec es una extensión al protocolo IP que proporciona seguridad a IP y a los protocolos de capas superiores. Fue desarrollado para el nuevo estándar IPv6 y después fue portado a IPv4. La arquitectura IPsec se describe en el RFC 2401[46].

Es un protocolo de seguridad "*de extremo a extremo*" toda la funcionalidad e inteligencia de la conexión VPN[47] reside en los puntos extremos; es decir, o en una pasarela (gateway) o en la computadora central Terminal. En los últimos tres años, IPSec ha sido el protocolo de tunelado IP predominante, y es actualmente la tecnología preferida a la hora de establecer conectividad de sitio a sitio por una red pública. En un principio, IPSec fue desarrollado para establecer comunicaciones privadas por redes IP públicas. El protocolo IPSec permite establecer dos funciones de seguridad principales:

- Autentificación, que permite asegurar la autenticidad e integridad del paquete IP completo.

- Encriptación, que permite asegurar la confidencialidad de los datos transportados.

El protocolo IPSec permite definir un túnel entre dos pasarelas. Una pasarela IPSec consistiría normalmente en un enrutador de acceso o un firewalls en el que esté implementado el protocolo IPSec. Las pasarelas IPSec están situadas entre la red privada del usuario y la red compartida del operador.

Los túneles IPSec se establecen dinámicamente y se liberan cuando no están en uso. Para establecer un túnel IPSec, dos pasarelas deben autentificarse y definir los algoritmos de seguridad y las claves que utilizarán para el túnel. El paquete IP original es encriptado en su totalidad e incorporado en encabezamientos de autentificación y encriptación IPSec. Se obtiene así la carga útil de un nuevo paquete IP cuyas direcciones IP de origen y destino son las direcciones IP de red pública de las pasarelas IPSec. Se establece así la separación lógica entre los flujos de tráfico de la VPN en una red IP

[46] *RFC 2401, "Arquitectura de Seguridad para IP"*
[47] *Virtual Private Network, puede consultar el capítulo 6 para obtener más información.*

compartida. Seguidamente, se utiliza un encaminamiento IP tradicional entre los extremos del túnel.

IPSec consigue estos objetivos mediante:

- Dos protocolos de seguridad de tráfico: el encabezamiento de autentificación (AH), que confiere integridad de los datos, y la carga útil de seguridad de encapsulación (ESP), que confiere integridad y confidencialidad de los datos.

- Un protocolo de gestión de clave criptográfica: el intercambio de claves por Internet (IKE), que se utiliza para negociar las conexiones IPSec.

IPSec está diseñado para proporcionar seguridad interoperable, de alta calidad, basada en criptografía para IPv4 e IPv6. El conjunto de servicios de seguridad que ofrece incluye:

- Control de acceso (previene uso no autorizado de recursos).
- Integridad sin conexión (detección de modificaciones en un datagrama IP individual).
- Autenticación del origen de los datos.
- Protección de la integridad de lo datos (detecta la llegada de datagramas IP duplicados).
- Confidencialidad (encriptación), y confidencialidad de flujos de trafico limitado.

Estos servicios se implementan en la capa IP, y ofrecen protección para éste nivel y a los niveles superiores. Estos Objetivos se llevan a cabo haciendo uso de dos protocolos de seguridad, la cabecera de autenticación (AH) y encapsulación de datos seguros, y a través de procedimientos de manejo de claves criptográficas y protocolos.

Los mecanismos utilizados están diseñados para ser independientes del algoritmo. Esta modularidad permite seleccionar diferentes conjuntos de algoritmos sin afectar a las otras partes de la implementación. La seguridad ofrecida depende en última instancia de la calidad de la implementación, y estará determinada por muchos factores (factor humano, físico, procedimientos, seguridad del Sistema Operativo).

Los servicios se proporcionan habilitando al sistema para elegir el protocolo de seguridad requerido, determinar el algoritmo para el servicio y haciendo uso de alguna clave criptográfica requerida para esos servicios. IPSec se puede utilizar para proteger una o más rutas entre un par de Hosts, un par de puertas de enlace seguras (sistemas intermedios entre el origen y el destino que implementan IPSec) o entre una puerta de enlace segura y un Host.

IPsec proporciona servicios de seguridad en la capa IP permitiendo a un sistema seleccionar los protocolos de seguridad, determinar los algoritmos a utilizar para los servicios, e implementar cualquier algoritmo criptográfico requerido para proporcionar los servicios solicitados.

5.1 Ventajas y limitaciones

El protocolo IPSec también proporciona las ventajas siguientes:

- Compatibilidad con la infraestructura de claves públicas**,** acepta el uso de certificados de claves públicas para la autenticación, con el fin de permitir relaciones de confianza y proteger la comunicación con hosts que no pertenezcan a un dominio en el que se confía.
- Compatibilidad con claves compartidas, si la autenticación mediante Kerberos[48] o certificados de claves públicas no es posible, se puede configurar una clave compartida (una contraseña secreta compartida) para proporcionar autenticación y confianza entre equipos.
- Transparencia de IPSec para los usuarios y las aplicaciones. Como IPSec opera al nivel de red, los usuarios y las aplicaciones no interactúan con IPSec.
- Administración centralizada y flexible de directivas mediante Directiva de grupo, cuando cada equipo inicia una sesión en el dominio, el equipo recibe automáticamente su directiva de seguridad, lo que evita tener que configurar cada equipo individualmente. Sin embargo, si un equipo tiene requisitos exclusivos o es independiente, se puede asignar una directiva de forma local.

[48] *Kerberos es un sistema de autenticación de seguridad estándar industrial indicado para ser distribuido bajo redes publicas.*

- Estándar abierto del sector, IPSec proporciona una alternativa de estándar industrial abierto ante las tecnologías de cifrado IP patentadas. Los administradores de la red aprovechan la interoperabilidad resultante.

El protocolo IPSec posee algunas limitantes, como las que se mencionan a continuación:

- No puede ser seguro si el sistema no lo es.
- No Provee seguridad en el nivel End-To-End (PGP, SSL, SSH)
- Autentifica máquinas y no a los usuarios.
- No detiene ataques DOS[49]
- No puede evitar el análisis de tráfico.
- El uso de IPSec requiere más tiempo de proceso en hosts y puertas de enlace que lo implementan.
- Requiere de mayor uso de memoria para el procesamiento de IPSec y su estructura de datos, así como el cálculo de los valores de comprobación de integridad, encriptación y desencriptación. Generando como consecuencia mayor tiempo de latencia.
- IPSec incrementa la utilización del ancho de banda en la transmisión debido al aumento del tamaño del paquete por la adición de las cabeceras AH y ESP.

5.2 Componentes de IPSec

La meta del protocolo IPsec es proporcionar varios servicios de seguridad para el tráfico de la capa IP, tanto a través de IPv4 e IPv6.
La estructura para el protocolo IP está definida y estandarizada por IETF, y el protocolo IPSec se encuentra en la RFC 2401[50], (Ver figura 47).

[49] *DoS (Denegación de Servicio). Un ataque DoS a un servidor conectado a Internet tiene como objetivo agotar sus recursos, ya sean de ancho de banda o de procesamiento, para que sea imposible acceder a él.*
[50] *RFC 2401, "Arquitectura de Seguridad para IP"*

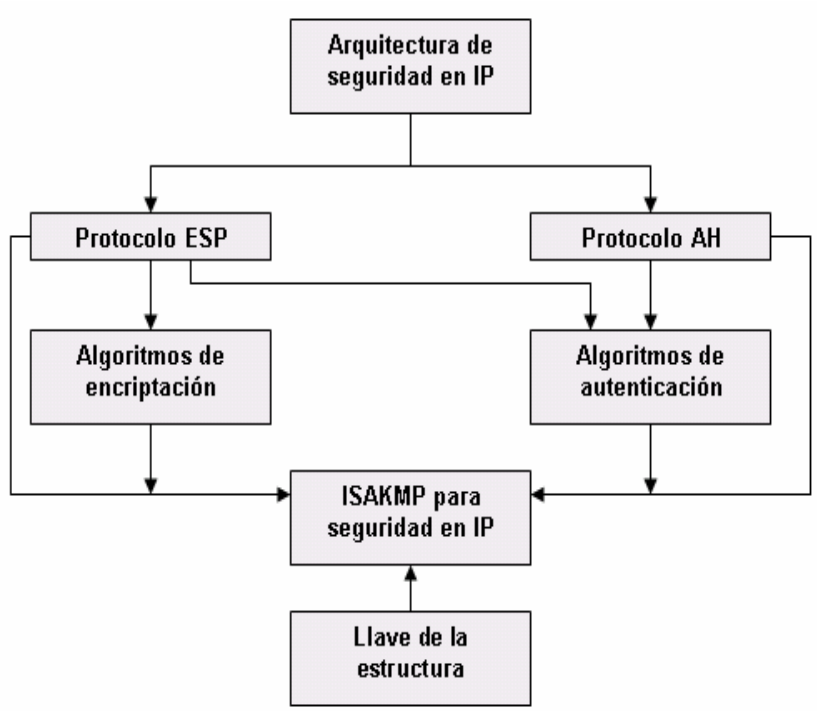


Figura 47. Relación entre los componentes de IP

Los componentes fundamentales de la arquitectura de seguridad IPSec son los siguientes:

1. Una general descripción de los requerimientos de seguridad y mecanismos que se encuentran en la capa de red.
2. Un elemento específico de seguridad utilizado para la encriptación (ESP, RFC 2406[51]).
3. Un elemento especifico de seguridad utilizado para la autenticación (AH, RFC 2402 [52])
4. Definiciones que son utilizadas por algoritmos de encriptación y autenticación.
5. Definiciones de políticas de seguridad y Asociaciones de Seguridad entre la comunicación de los nodos.
6. Llaves para manejar la información de IPsec.

[51] *RFC 2406, "Encriptación de datos en IP (ESP)"*

[52] *RFC 2402, " Cabecera de Autenticación IP"*

Se debe de especificar que éstos mecanismos son considerados genéricos, es decir que ellos pueden ser utilizados para contextos de IPv4 e IPv6. Estos protocolos pueden existir como software adicional para IPv4, en cambio, para IPv6 se encuentran integrados. La forma de operación del protocolo IPsec es similar a la de otros protocolos de seguridad como IPSP (IP Security Policy)[53] y X.509[54] .

5.2.1 Características de IPSec

Las siguientes características de IPSec afrontan los siguientes métodos de ataque:

- Protocolo Carga de seguridad de encapsulación (ESP, Encapsulating Security Payload). ESP proporciona privacidad a los datos mediante el cifrado de los paquetes IP.

- Claves basadas en criptografía. Las claves cifradas, que se comparten entre los sistemas que se comunican, crean una suma de comprobación digital para cada paquete IP. Cualquier modificación del paquete altera la suma de comprobación, mostrando al destinatario que el paquete ha sido cambiado en su tránsito. Se utiliza material de claves diferente para cada segmento del esquema de protección global y se puede generar nuevo material de claves con la frecuencia especificada en la directiva de IPSec.

- Administración automática de claves. Las claves extensas y el cambio dinámico de claves durante las comunicaciones ya establecidas protegen contra los ataques. IPSec usa el protocolo Asociación de seguridad en Internet y administración de claves (ISAKMP, Internet Security Association and Key Management Protocol) para intercambiar y administrar dinámicamente claves cifradas entre los equipos que se comunican.

[53] *IPSP, políticas de seguridad para el protocolo IP.*
[54] *X.509. es un estándar ITU para infraestructura de claves pública especifica los formatos para certificados de claves públicas y un algoritmo de validación de ruta de certificación*

- Negociación de seguridad automática. IPSec usa *ISAKMP* para negociar de forma dinámica un conjunto de requisitos de seguridad mutuos entre los equipos que se comunican. No es necesario que los equipos tengan directivas idénticas, sólo una directiva configurada con las opciones de negociación necesarias para establecer un conjunto de requisitos con otro equipo.

- Seguridad a nivel de red. IPSec existe en el nivel de red, proporcionando seguridad automática a todas las aplicaciones.

- Autenticación mutua. IPSec permite el intercambio y la comprobación de identidades sin exponer la información a la interpretación de un atacante. La comprobación mutua (autenticación) se utiliza para establecer la confianza entre los sistemas que se comunican. Sólo los sistemas de confianza se pueden comunicar entre sí. Los usuarios no tienen que estar en el mismo dominio para comunicar con la protección de IPSec. Pueden estar en cualquier dominio de confianza de la empresa. La comunicación se cifra, lo que dificulta la identificación e interpretación de la información.

5.3 Cabeceras y algoritmos.

IPSec permite el tráfico seguro haciendo uso de dos protocolos:

1. La **cabecera de autenticación** (AH), que proporciona integridad sin conexión, autenticación del origen de datos y un servicio de integridad de datos.
2. El **protocolo de encapsulado de datos seguros** (ESP), proporciona confidencialidad (encriptación) y confidencialidad de flujo de tráfico limitado. También puede brindar integridad sin conexión, autenticación del origen de datos.

AH y ESP son acceso de control basados en la distribución de clave criptográfica y la administración de flujo de tráfico vinculada a estos protocolos de seguridad. Ambos protocolos se pueden utilizar individualmente o combinados para proporcionar los servicios de seguridad deseados en IPv4 e IPv6. Cada protocolo permite dos modos de uso, *modo de transporte y modo de túnel*. El primero brinda protección principalmente a

los protocolos de las capas superiores, mientras que el modo túnel se utiliza para paquetes IP bajo túneles.

Algoritmos de Autenticación.

Una implementación adecuada de la cabecera AH tiene que soportar obligatoriamente los algoritmos:

- HMAC con MD5.
- HMAC con SHA-1.

Para proteger la confidencialidad de lo datagramas IP, los protocolos IPsec emplean algoritmos estándar de cifrado simétrico. El estándar IPsec exige la implementación de NULL y DES. En la actualidad se suelen emplear algoritmos más fuertes: 3DES, AES.

5.4 Modos túnel y transporte

IPsec emplea dos protocolos diferentes (AH y ESP) para asegurar la autenticación, integridad y confidencialidad de la comunicación. Puede proteger el datagrama IP completo o sólo los protocolos de capas superiores. Estos modos se denominan, respectivamente, modo túnel y modo transporte. En modo túnel el datagrama IP se encapsula completamente dentro de un nuevo datagrama IP que emplea el protocolo IPsec. En modo transporte IPsec sólo maneja la carga del datagrama IP, insertándose la cabecera IPsec entre la cabecera IP y la cabecera del protocolo de capas superiores (vea la figura 48).

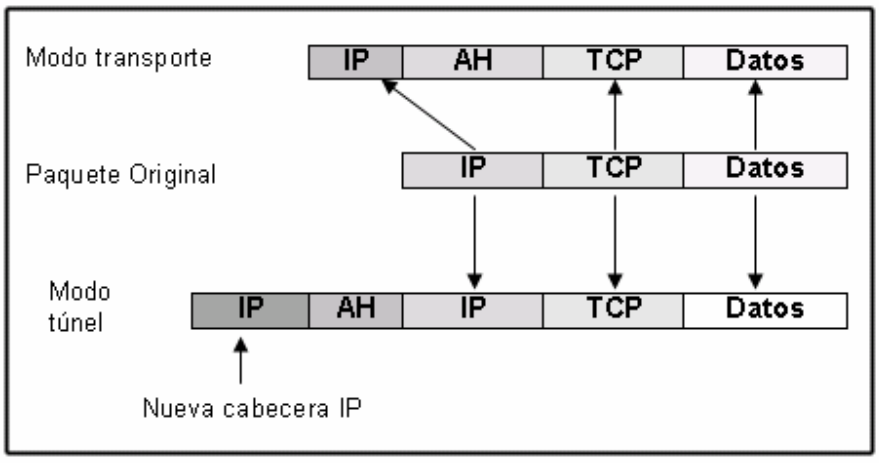


Figura 48 IPsec en modo transporte y túnel.

5.5 El protocolo ISAKMP

El protocolo ISAKMP (Internet Security Association Key Management Protocol), se selecciona para el intercambio de claves y parámetros de seguridad en IPSec. ISAKMP es un protocolo que proporciona la infraestructura necesaria para la negociación de asociaciones de seguridad (SA) entre dos usuarios cualesquiera.

Se define como transacción de configuración como un doble intercambio dónde el emisor realiza un envío-petición (Set - Request) y el receptor contesta mediante un reconocimiento de petición-respuesta (Acknowledge-Reply). De esta forma a un envío (Set) le corresponde un reconocimiento de envío (Acknowledge) y a una petición (Request) una respuesta (Reply). (Ver figura 49).

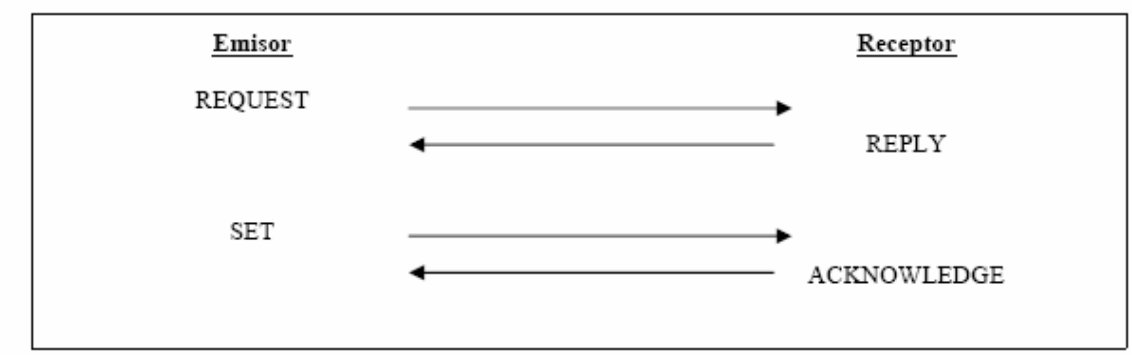


Figura 49. Comunicación entre el Emisor y Receptor

Este esquema permite evitar ataques de negación de servicio (DOS); ya que hasta que no se reciba la respuesta el esquema no continúa. Posteriormente se producen los intercambios de información necesarios mediante envíos-reconocimientos de envío (Set - Acknowledge) dónde se negocian los diferentes parámetros de seguridad que gobernaran la comunicación. El intercambio de mensajes mediante ISAKMP se realiza mediante el esquema de cabeceras de extensión (ver figura 50) el cual es utilizado en la definición del protocolo IP6.

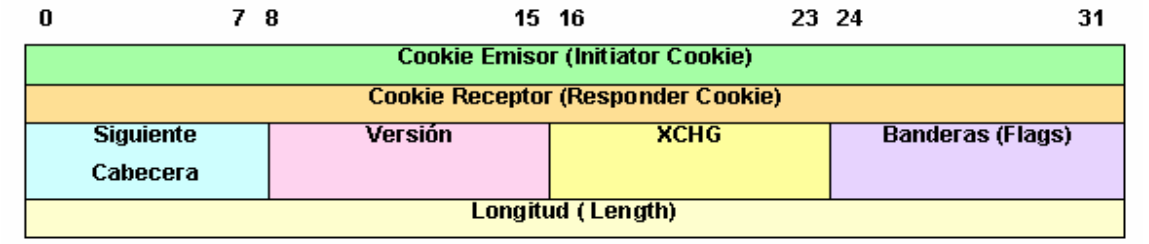


Figura 50. Formato de la cabecera del ISAKMP.

El intercambio de claves entre el emisor y el receptor se realiza utilizando el algoritmo de Diffie-Hellman.

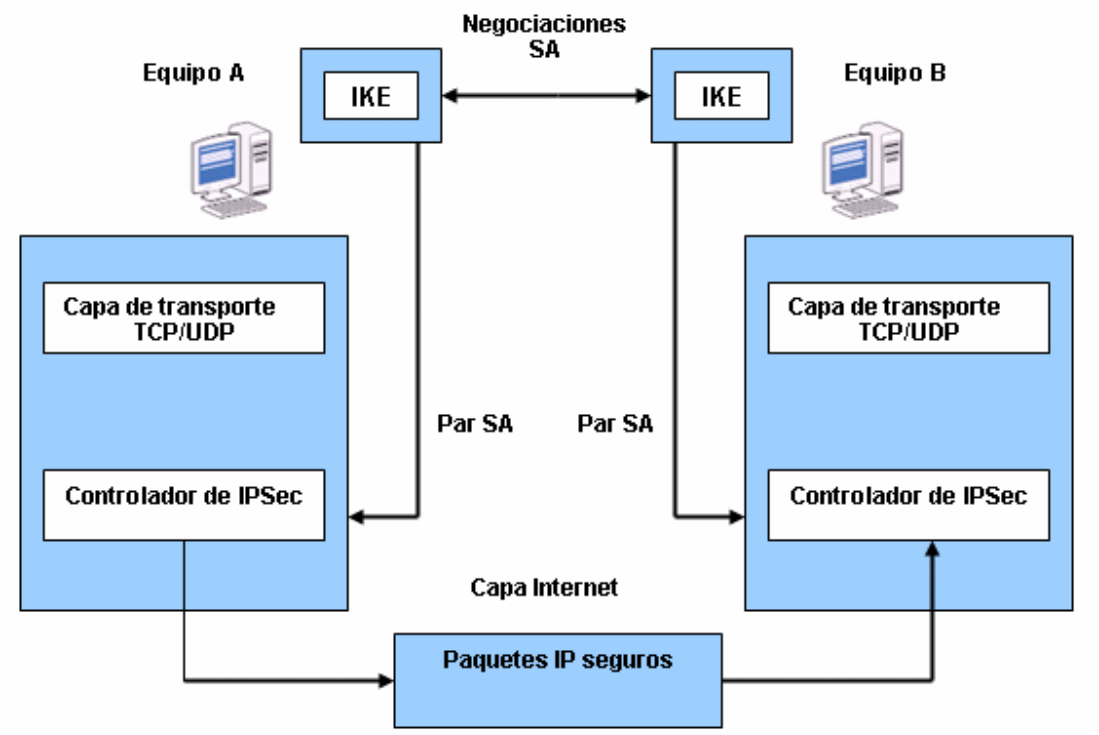
5.6 Internet Key Exchange (IKE)

El protocolo IKE resuelve el problema más importante del establecimiento de comunicaciones seguras: *la autenticación de los participantes y el intercambio de claves simétricas*. El protocolo IKE suele implementarse a través de servidores de espacio de usuario, y no suele implementarse en el sistema operativo. El protocolo IKE emplea el puerto 500 UDP para su comunicación.

El protocolo IKE está diseñado para establecer de manera segura una relación de confianza entre dos equipos, negociar las opciones de seguridad y generar de manera dinámica el material de las claves criptográficas secretas compartidas. La configuración del acuerdo de seguridad asociado al material de claves se denomina asociación de seguridad o SA. Estas claves proporcionarán autenticidad, integridad y, opcionalmente, cifrado de los paquetes IP que se envían mediante la asociación de seguridad. IKE negocia dos tipos de asociaciones de seguridad:

- Una asociación de seguridad de modo principal (la asociación de seguridad IKE que se utiliza para proteger la propia negociación IKE).
- Asociaciones de seguridad IPSec (las asociaciones de seguridad que se utilizan para proteger el tráfico de la aplicación).

El filtro de paquetes se interpreta de dos formas: una utiliza sólo la dirección y la información de identidad para permitir a IKE establecer una asociación de seguridad de modo principal (la asociación de seguridad IKE); la otra permite a IKE establecer las asociaciones de seguridad IPSec (también conocidas como asociaciones de seguridad de



modo rápido). (Ver figura 51).

Figura 51. Funcionamiento del protocolo IKE

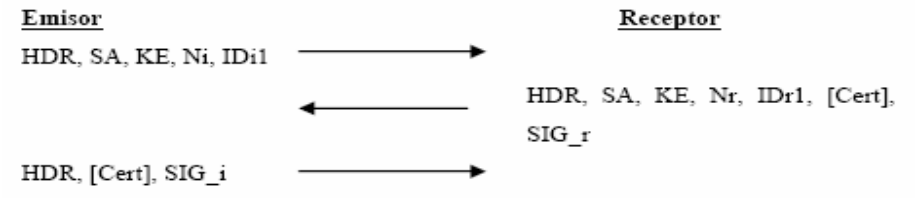
5.6.1 Fases de IKE

El protocolo IKE es un protocolo de dos fases para el establecimiento de un canal auténtico y seguro entre dos usuarios. Este protocolo utiliza la infraestructura de mensajes del protocolo ISAKMP para el intercambio de mensajes.

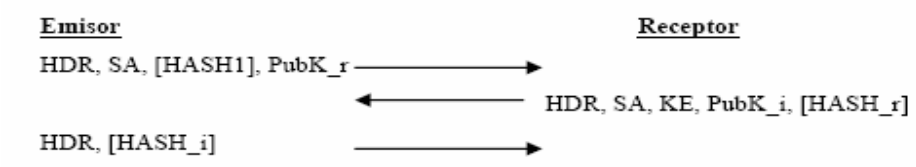
Fase1: Se negocian las asociaciones de seguridad (SA), se utiliza el protocolo Diffie-Hellman para el intercambio de una clave común y se establece el algoritmo de cifrado (3DES[55]), el algoritmo de Hash (MD5[56]) y del sistema de autenticación. En esta fase tanto el emisor como el receptor quedan autenticados mediante uno de los siguientes cuatro métodos:

1. Autenticación con claves pre-compartidas (Pre-shared Keys).

2. Autenticación mediante firmas digitales (Digital Signatures).

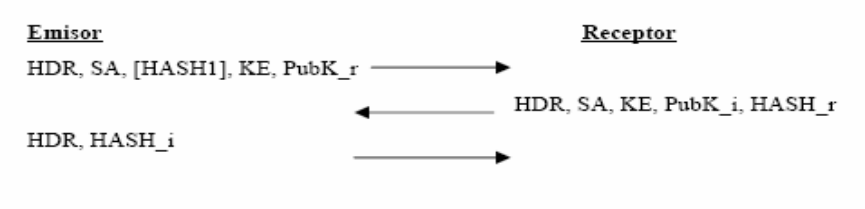


3. Autenticación mediante clave pública 1.



[55] *3DES, se llama al algoritmo que hace triple cifrado del DES. También es conocido como TDES.*
[56] *Message-Digest Algorithm 5, (Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits.*

4. Autenticación mediante clave pública 2.



Donde:

HDR: Cabecera ISAKMP.	HDR*: Cabecera cifrada
HASH: Función Hash.	KE: Valor público Diffie-Hellman.
SA: Asociación de seguridad.	Ni y Nr: Valor temporal (Nonce payload).
[Cert] y SIG: Certificado y firma digital.	ID: Identificador.
PubK: Clave pública.	[ ] : Opcional.

Fase 2: Una vez establecidos los distintos parámetros iniciales (SA) y aprovechando la seguridad de la fase 1, se inicia un modo rápido (Quick Mode) dónde se vuelven a negociar asociaciones de seguridad (SA) con el objetivo de evitar ataques de reutilización (Replay) de los datagramas de la fase 1 por un atacante.

5.7 Procesamiento de los paquetes de entrada y salida.

a) Procesamiento de los Paquetes de Salida.

Los paquetas salientes se han de preparar si se requiere el uso de la AH.

1. **Búsqueda de la Asociación de Seguridad (SA):** Sólo si la implementación de IPSec indica que un paquete de salida está asociado a una SA se aplicará la cabecera AH.

2. **Generación del Número de Secuencia**: Se debe de indicar el número de secuencia, no se puede enviar un número que obligue a rotar el número de Secuencia (es decir, que tome el valor de cero).

3. **Cálculo del ICV**: El valor de comprobación de integridad de la cabecera AH se calcula sobre:
   - Los campos de la cabecera IP que son inmutables, o predecibles por el receptor.
   - La cabecera AH: "Next header", longitud de la carga, el campo reservado, número de secuencia, y los datos de autenticación, (que tienen el valor de cero para este cálculo), y todos los bytes de relleno explícitos si los hay.
   - Los datos del protocolo de capa superior, que son inmutables durante el tránsito.

b) *Procesamiento de los Paquetes de Entrada.*

Si existe más de una cabecera o extensión IPSec, el procesamiento de una cabecera ignora a las siguientes. Si es requerido se hará un reensamblado de los paquetes antes del procesamiento de la cabecera AH.

1. **Búsqueda de la Asociación de Seguridad:** cuando se recibe un paquete que contiene una AH se determina a que SA pertenece. Si no existe una SA válida el receptor tiene que descartar el paquete.

2. **Verificación del Número de Secuencia:** todas las implementaciones con la cabecera AH tienen que soportar el servicio anti-replay. Aunque el receptor puede habilitar o no su uso. Si el receptor tiene habilitado el servicio anti-replay para la SA, el contador de paquetes recibidos tiene que inicializarse a 0 cuando se establece la SA. El receptor debe verificar si el paquete recibido contiene un número de secuencia que no duplique a otro número de secuencia de los paquetes recibidos durante el tiempo de vida de la SA. Este debe ser el primer campo que se verifique después de que se ha comprobado que pertenece a una SA, para acelerar el rechazo de paquetes duplicados.

3. **Verificación del ICV:** El receptor calcula el ICV sobre los campos del paquete apropiados, usando el algoritmo de integridad especificado, y verifica que el ICV obtenido coincide con el enviado en el paquete.
   Si el ICV calculado y el recibido coinciden, entonces el datagrama es válido y el paquete será aceptado. Si no coinciden el paquete será descartado, pudiéndose auditar este evento.

5.8 El proceso de negociación y filtrado

Cuando un equipo configurado con una directiva de IPSec intenta comunicarse con otro equipo, comienza el proceso siguiente:

1. Las directivas de IPSec se entregan al controlador de IPSec y el intercambio de clave ISAKMP a través de directivas locales.
2. ISAKMP supervisa las negociaciones entre los hosts y proporciona claves que se utilizan con algoritmos de seguridad.
3. El controlador de IPSec supervisa, filtra y protege el tráfico entre el nivel de transporte y el nivel de red.

5.9 Directivas de seguridad de IP

Las directivas son las reglas de seguridad que definen el nivel de seguridad deseado, el algoritmo de hash, el algoritmo de cifrado y la longitud de la clave. Estas reglas también definen las direcciones, protocolos, nombres DNS, subredes o tipos de conexión a los que se aplica la configuración de seguridad. Las directivas de IPSec se pueden configurar de acuerdo con los requisitos de seguridad de un usuario, grupo, aplicación, dominio, sitio o empresa global.

5.9.1 ISAKMP y directivas de seguridad

Durante la configuración de IPSec, se crea una directiva en la interfaz. Sin embargo, IPSec crea las dos siguientes directivas de negociación de seguridad en segundo plano:

- La primera negociación incluye autenticación de identidad de usuario para los dos hosts que se van a comunicar y el intercambio de las claves de la sesión para proteger los datos. ISAKMP administra esta primera negociación, que se puede llamar directiva de negociación.
- La segunda negociación sigue al intercambio de las claves. Los dos hosts tienen que acordar la configuración de seguridad que van a utilizar para proteger la comunicación sobre IP. A la directiva que define las reglas de esta negociación se le llama *directiva de seguridad*.

5.9.2 Componentes de seguridad

Proporcionan la capacidad para iniciar y controlar una comunicación segura en función del origen, el destino y el tipo de tráfico IP; una o todas pueden estar activas de forma simultánea. Se adaptan a una amplia gama de comunicaciones entre cliente y servidor.

Componentes:

1. Métodos de seguridad. Especifica cómo los equipos que se comunican tienen que proteger el intercambio de datos. Puede utilizar los métodos predefinidos Medio y Alto, o definir métodos de seguridad personalizados.

2. Configuración de túneles. En algunas situaciones, como entre routers que sólo están conectados por Internet, debe considerar habilitar el modo de túnel en IPSec. El extremo final del túnel es el equipo del túnel más próximo al destino del tráfico IP, como se especifica en la lista del filtro asociado. Para definir un túnel IPSec tiene que haber dos reglas, una para cada sentido.

3. Métodos de autenticación. El método de autenticación define cómo cada usuario se asegurara de que el otro equipo o el otro usuario son realmente quienes dicen ser, por ejemplo:

- Kerberos: el protocolo de seguridad Kerberos es la tecnología de autenticación predeterminada. Este método se puede usar en cualquier cliente que ejecute el protocolo Kerberos.
- Certificados: este método requiere que se haya configurado al menos una entidad emisora de certificados como por ejemplo X.509.
- Clave previamente compartida: es una clave secreta, compartida, que dos usuarios acuerdan de antemano y que configuran manualmente antes de usarla. Cada regla puede estar configurada con uno o varios.

Capítulo

6

Filtros de seguridad.

Temas:

- Filtros.
- ICMPv6.
- Tipos de filtros.

6. Filtros

En el área de seguridad de redes, es necesario destacar la importancia del empleo de los filtros de paquetes, su función primordial es brindar seguridad sobre la información que circula en la red sea la correcta y deseada. Los filtros de los paquetes son la base de la construcción de Firewalls, los cuales brindan protección a los usuarios ante los ataques de muchos intrusos. Los filtros también se pueden aplicar al protocolo IPv6, en este capítulo se aborda el filtrado de paquetes desde tres perspectivas: ICMPV6, Listas de control de acceso (ACL), ip6tables.

El filtrado de paquetes es un mecanismo de inspección, manejo y control de los paquetes; verifican la entrada y salida en una red determinada y toman decisiones sobre otorgar permiso para que atraviesen la red o descartarlos. Se utilizan para la implementación de Firewalls, control de acceso, Interconexión de redes e Implementación de políticas de uso. Un filtro puede tomar tres decisiones sobre un paquete, aceptarlo, rechazarlo o descartarlo.

6.1 Motivos de un filtro

Los motivos básicos para establecer un filtro son la regulación y control del tráfico de un host o red. Los objetivos de establecer filtros se pueden resumir en dos características: seguridad y rendimiento. Seguridad porque se decide el acceso a los servicios y rendimiento porque se mejoraran las prestaciones de la red y reduce el tráfico innecesario.

- La regulación es la decisión que se debe de tomar con respecto al tráfico que se permite y el que se prohíbe en función de los orígenes y destinos de los paquetes circulantes.
- El control es la posibilidad de analizar y manipular las cabeceras de los paquetes para que se adapten a las necesidades.

6.1.1 Reglas

Los filtros funcionan mediante reglas o criterios para seleccionar paquetes, las cuales forman grupos, y de forma secuencial verifican si los paquetes concuerdan con los criterios establecidos de selección. Las reglas utilizadas para filtras paquetes pueden ser estáticas o dinámicas. Los criterios de selección de paquetes pueden ser entre otros:

- Protocolos (TCP[57], UDP[58], ICMP[59])
- Direcciones de origen
- Direcciones de destino
- Puertos (UDP, TCP)
- Interfaz
- Banderas de TCP (SYN, ACK)
- Tipo del paquete (ICMP)

6.2 ICMPv6

El protocolo de Mensajes de Control de Internet (ICMP), descrito originalmente en el documento RFC792[60] para IPv4, ha sido actualizado para permitir su uso bajo IPv6 con la finalidad de reportar mensajes de errores ocurridos durante el procesamiento de los paquetes y efectuar análisis del estado de la red.

ICMPv6 posee un valor para el campo de siguiente cabecera y éste es igual a 58. Es parte integral de IPv6 y debe ser totalmente incorporado en cualquier implementación de un host IPv6; se emplea también para la realización de otras funciones relativas, como por ejemplo diagnósticos a través de los comandos ping y tracert.

---

[57] *Protocolo de Control de Transmisión, garantiza los datos que serán entregados en su destino sin errores y en el mismo orden en que se transmitieron*

[58] *User Datagram Protocol, permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión.*

[59] *Protocolo de Control de Mensajes de Internet, subprotocolo de diagnóstico y notificación de errores*

[60] *RFC 792,"protocolo de mensajes de control Internet"*

- **Ping:** se utiliza para enviar mensajes de solicitud de eco Icmpv6 y registrar la recepción de los mensajes de respuesta de eco ICMPv6. Mediante ping, se detectan errores de comunicación en la red o en los hosts y ayuda a solucionar problemas comunes de conectividad IPv6.
- **Tracert**: se emplea para enviar mensajes de solicitud de eco ICMPv6 con valores de incremento gradual en el campo Hop Limit (límite de saltos). Tracert traza y muestra la ruta seguida por los paquetes IPv6 entre un origen y un destino, con lo que se puede solucionar problemas comunes de enrutamiento IPv6.

El protocolo ICMPv6 proporciona también un marco de trabajo para los protocolos MLD y ND los cuales se detallan a continuación. Estos protocolos marcan diferencias entre el protocolo ICMP para IPv4 e ICMPv6 (Ver figura 52)

1. Descubrimiento de escucha de multidifusión (MLD)

MLD consiste en una serie de mensajes ICMPv6 que reemplazan la versión 2 del Protocolo de administración del grupo Internet IGMP[61] para IPv4 en la administración de la pertenencia a multidifusión[62] de subred.

2. Descubrimiento de vecinos (ND)

El descubrimiento de vecinos consiste en una serie de mensajes ICMPv6 que administran la comunicación de un host a otro en un vínculo, y determina las direcciones de nivel 2 de otros hosts en el mismo enlace. El ND reemplaza al Protocolo de resolución de direcciones ARP[63]. El proceso de detección de vecinos se encuentra detallado en la RFC 2461[64].

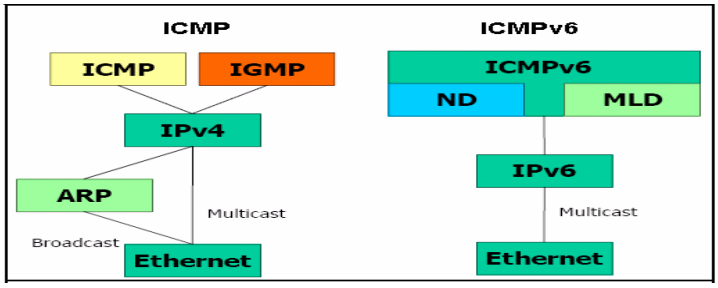


Figura 52 Comparación entre ICM e ICMPv6.

[61] *IGMP (Internet Group Management Protocol), se utiliza para intercambiar información acerca del estado de pertenencia entre enrutadores IP que admiten la multidifusión y miembros de grupos de multidifusión.*
[62] *El tráfico de multidifusión se envía a una única dirección, pero se procesa en múltiples hosts.*
[63] *Address Resolution Protocol (Protocolo de resolución de direcciones); es responsable de encontrar la dirección hardware (Ethernet MAC) que corresponde a una determinada dirección IP.*
[64] *RFC 2461, "Descubrimiento del Vecindario para IPv6 (ND)"*

6.2.1 Formato de la cabecera ICMPv6

El formato de la cabecera ICMPv6 esta formada por los campos: tipo, código, checksum, información y el cuerpo del mensaje. (Ver figura 53)

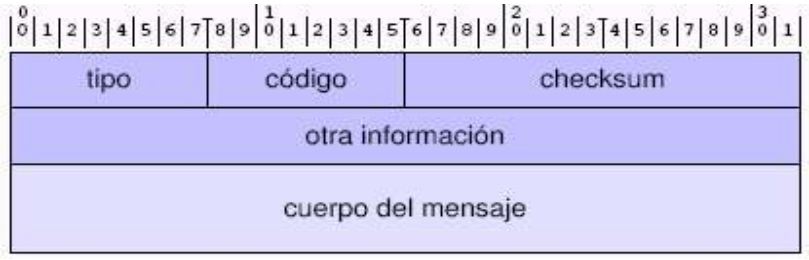


Figura 53. Formato de la cabecera ICMPv6

- **Tipo**: indica el tipo de mensaje, y su valor determina el formato 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 o código de redundancia**: permite detectar errores en el mensaje ICMPv6.
- **Cuerpo del mensaje**: este campo de de tamaño variable, contiene datos diferentes según el tipo de mensaje. El tamaño total del paquete ICMPv6 no debe de exceder del valor mínimo MTU de IPv6, es decir de 1280 bytes.

6.2.2 Mensajes ICMPv6

Los mensajes ICMPv6 se agrupan en dos tipos o clases: mensajes de error, y mensajes informativos.

6.2.2.1 Mensajes de error

Tienen cero en el bit de mayor peso del campo tipo, por lo tanto sus valores se sitúan entre 0 y 127; los tipos de errores se muestran a continuación (Ver tabla 6):

- **Destino no alcanzable (destination unreachable):** se genera cuando un enrutador no puede entregar un datagrama, estos mensajes se envían a la dirección fuente del paquete. El campo de datos contiene parte del paquete que ha generado el error. Si el destino no está alcanzable por congestión no se generará ningún mensaje ICMP.

- **Packet Too Big (paquete demasiado grande):** se originan cuando un enrutador no puede encaminar un datagrama porque su longitud es más grande que el MTU del enlace.
- **Time Exceeded (tiempo agotado):** se generan cuando el Hop Limit de un datagrama contiene el valor de cero, o cuando se agota el tiempo permitido para reunir los fragmentos.
- **Parameter Problem (problema de parámetros)**: el mensaje se crea cuando un host no consigue procesar un paquete debido a problemas en la identificación de los campos de la cabecera; el host rechaza el paquete y envía un mensaje ICMP "Parameter Problem" a la fuente.

Mensajes de error ICMP		
Tipo	**Descripción y Códigos**	
1	**Destino no alcanzable (Destinación Unreachable)**	
	Código	Descripción
	0	Sin ruta hacia el destino
	1	Comunicación prohibida administrativamente
	2	Sin Asignar
	3	Dirección no alcanzable
	4	Puerto no alcanzable
2	**Paquete demasiado grande (Packet too Big)**	
3	**Tiempo excedido (Time Exceded)**	
	Código	Descripción
	0	Limite de saltos excedido
	1	Tiempo de desfragmentación excedido
4	**Problema de parámetros (Parameter Problem)**	
	0	Campo erróneo en cabecera
	1	Tipo de "cabecera siguiente " desconocida
	2	Opción IPv6 desconocida

Tabla 6 Mensajes de error ICMPv6.

6.2.2.2 Mensajes de información

La RFC 2463[65] define dos tipos de mensajes ICMP: Echo Request y Echo Reply. Existen otros tipos que se utilizan para el proceso de detección de vecinos (Neighbor Discovery), descubrimiento de MTU y gestión de grupos Multicast. Los mensajes ICMP Echo Request y Echo Reply pueden ser autenticados por una cabecera de tipo Authentication Header, para que el destino pueda comprobar el origen de las peticiones y rechazar los mensajes no confiables. Los valores de los mensajes informativos oscilan entre 128 y 255; los tipos de errores se muestran a continuación (Ver tabla 7):

- **Echo Request:** el mensaje de petición de eco tiene un tipo de 128 y el código puesto a cero. Los campos identificación y número de secuencia permiten acoplar peticiones con respuestas. Los datos que se incluyen en los mensajes ICMP dependen de la implementación, y desde ellos es posible identificar el tipo de host que los envía.

- **Echo Reply:** el mensaje de respuesta de eco tiene un tipo de 129 y el código puesto a cero. Los campos identificación y número de secuencia deben ser los mismos que los de la petición correspondiente, Los datos se copian del mensaje de petición sin modificar.

Mensajes Informativos ICMPv6		
Tipo	**Descripción y Códigos**	
128	Solicitud de eco (Echo Request)	
129	Respuesta de eco (Echo Reply)	
130	Group Membership Quero (MLD)	
	Código	Descripción
	1	General Quero
	2	Multicast Address Specific Query
131	Multicast Address Report	
132	Multicast Address Done	

Tabla 7 Mensajes informativos ICMPv6

[65] *RFC 2463, "Protocolo de Mensajes de Control de Internet para IPv6 (ICMPv6)"*

6.3 Tipos de filtros

a. Filtros a nivel de paquete:

Esta tecnología pertenece a la primera generación; la cual analiza el tráfico de la red. Cada paquete que entra o sale de la red es inspeccionado y lo acepta o rechaza basándose en las reglas definidas por el usuario. El filtrado de paquetes es efectivo y transparente para los usuarios de la red, pero es difícil de configurar..

Las reglas para rechazar o aceptar un paquete son las siguientes:

• Si no se encuentra una regla definida y establecida para aplicar al paquete, entonces el paquete es rechazado.

• Si se encuentra una regla que se aplica al paquete, y la regla permite el paso, se establece la comunicación.

• Si se encuentra una regla que se aplica al paquete, y la regla rechaza el paso, el paquete es rechazado.

b) Firewall a nivel circuito (Circuit Level Firewalls):

Aplica mecanismos de seguridad cuando una conexión TCP o UDP es establecida. Una vez que la conexión se establece, los paquetes pueden ir y venir entre las computadoras sin tener que ser revisados cada vez. Mantiene una tabla de conexiones válidas y permite que los paquetes de la red pasen a través de ella si corresponden a algún registro de la tabla. Una vez terminada la conexión, la tabla se borra y la transmisión de información entre las dos computadoras se cierra.

c) A nivel aplicación:

Examina la información de todos los paquetes de la red y mantiene el estado de la conexión y la secuencia de la información. En este tipo de tecnología también se puede validar claves de acceso y algunos tipos de solicitudes de servicios. La mayoría de estos tipos requieren de software especializado y servicios Proxy. Un Servicio Proxy es un programa que aplica mecanismos de seguridad a ciertas aplicaciones, tales como FTP o HTTP. Un servicio proxy puede incrementar el control al acceso, realizar chequeos detallados a los datos y generar auditorias sobre la información que se transmite.

Capítulo

7

Firewall.

Temas:

- Firewall.
- Funciones principales de un Firewall.
- Que puede realizar un Firewall.
- Tipos de firewall.
- Tecnologías de los firewalls.
- Firewalls basados en Linux.
- Netfilter6.
- Ip6tables.

7. Firewall

Un Firewall es un arreglo de hardware / software especifico con un sistema operativo o una IOS que filtra el tráfico TCP/UDP/ICMP/../IP y decide si un paquete pasa, se modifica, se convierte o se descarta. Para que un Firewall entre redes funcione como tal debe tener al menos dos tarjetas de red. Esta sería la tipología clásica de un Firewall:

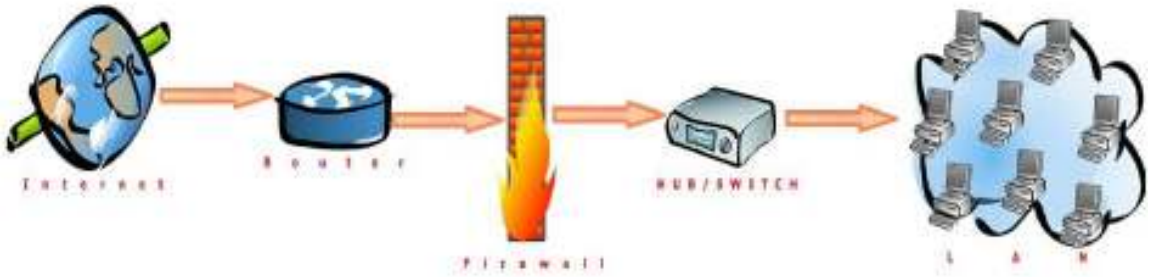


Figura 54 Esquema de Firewall típico entre red local e Internet

Esquema típico de Firewall para proteger una red local conectada a Internet a través de un router. El Firewall debe colocarse entre el router (con un único cable) y la red local (conectado al switch o al hub de la LAN)

7.1 Tipos de Firewall

7.1.1 Firewall de capa de red

El primero funciona al nivel de la red de la pila de protocolos (TCP/IP) como filtro de paquetes IP, no permitiendo que estos pasen el Firewall a menos que se atengan a las reglas definidas por el administrador del Firewall o aplicadas por defecto como en algunos sistemas inflexibles de Firewall. Una disposición más permisiva podría permitir que

cualquier paquete pase el filtro mientras que no cumpla con ninguna regla negativa de rechazo.

7.1.2 Firewall de capa de aplicación

El segundo trabaja en el nivel de aplicación, todo el tráfico de HTTP, (u otro protocolo), puede interceptar todos los paquetes que llegan o salen de una aplicación. Se bloquean otros paquetes (generalmente sin avisar al remitente). En principio, los Firewalls de aplicación pueden evitar que todo el tráfico externo indeseado alcance las máquinas protegidas.

7.2 Ventajas de un Firewall

- Protege de intrusiones.- Solamente entran a la red las personas autorizadas basadas en la política de la red basándose en las configuraciones.
- Optimización de acceso.- Identifica los elementos de la red internos y optimiza que la comunicación entre ellos sea más directa si así se desea. Esto ayuda a reconfigurar rápida y fácilmente los parámetros de seguridad.
- Protección de información privada.- Permite el acceso solamente a quien tenga privilegios a la información de cierta área o sector de la red.
- Protección contra virus.
- Evita que la red se vea infestada por nuevos virus que
- sean liberados.
- Los Firewalls tienen a menudo funcionalidad de traducción de direcciones de red (NAT) y es común utilizar el así llamado espacio de direcciones privado en las máquinas detrás de ella. Este espacio de direcciones privado se realiza como un intento (de eficacia discutible) de disfrazar las direcciones internas o red.
- La configuración correcta de Firewalls se basa en conocimientos considerables de los protocolos de red y de la seguridad de la computadora. Errores pequeños pueden dejar a un Firewall sin valor como herramienta de seguridad.
- Un Firewall es un dispositivo que filtra el tráfico entre redes, como mínimo dos.

- El Firewall puede ser un dispositivo físico o un software sobre un sistema operativo. En general debemos verlo como una caja con DOS o mas interfaces de red en la que se establecen una reglas de filtrado con las que se decide si una conexión determinada puede establecerse o no. Incluso puede ir más allá y realizar modificaciones sobre las comunicaciones, como el NAT.

7.3 Firewalls basados en Linux

Originalmente los firewalls elaborados por medio del sistema operativo Linux se basaban en el código **ipfw**. Este código comprende la versión original de los firewalls construidos con el kernel de Linux.

Un firewall es un hardware específico con un sistema operativo o una IOS que filtra el tráfico TCP, UDP, ICMP, IP y decide si un paquete pasa, se modifica, se convierte o se descarta. Para que un firewall entre redes funcione como tal debe tener al menos dostarjetas de red. La siguiente figura muestra una topología clásica de un firewall:

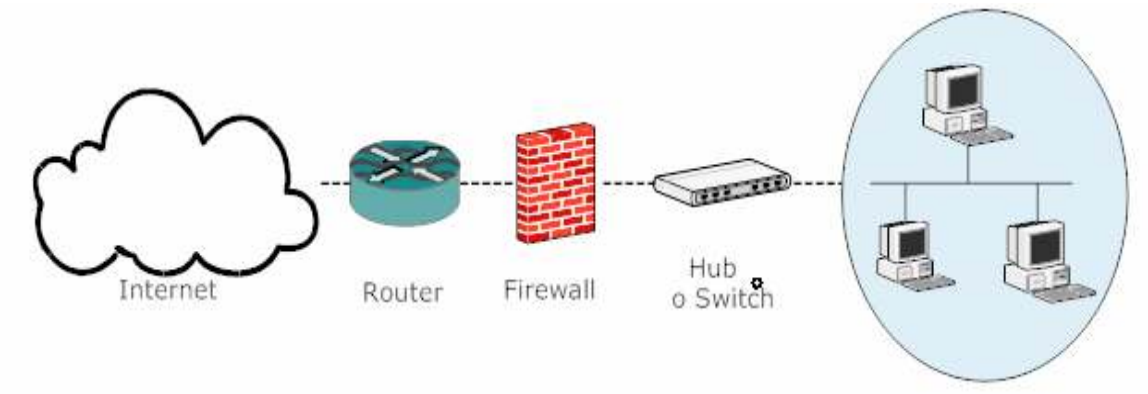


Figura 55 Esquema de un firewall típico entre red local e Internet.

En la figura anterior se muestra un esquema típico de firewall para proteger una red local conectada a Internet a través de un router. El firewall debe colocarse entre el router (con un único cable) y la red local (conectado al switch o al hub de la LAN)

Dependiendo de las necesidades de cada red, puede colocarse uno o más firewalls para establecer distintos perímetros de seguridad en torno a un sistema. Es frecuente también que se necesite exponer algún servidor a Internet (como es el caso de un servidor Web, un servidor de correo, etc.). En esa situación se debe de situar ese servidor en lugar

aparte de la red, el que denominamos DMZ o zona desmilitarizada. El firewall tiene entonces tres entradas:

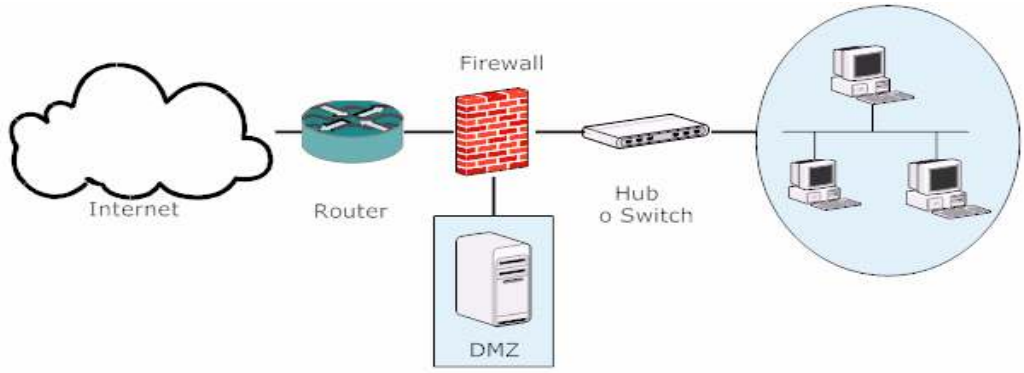


Figura 56 Esquema de firewall entre la red local e Internet con zona DMZ para servidores expuestos.

En la zona desmilitarizada se pueden colocar muchos servidores como se necesiten. Con esta arquitectura, se permite que el servidor sea accesible desde Internet de tal forma que si es atacado y se obtiene acceso a él, la red local sigue protegida por el Firewall. La estructura de DMZ puede realizarse también con doble firewall, así como se visualiza en la siguiente figura.

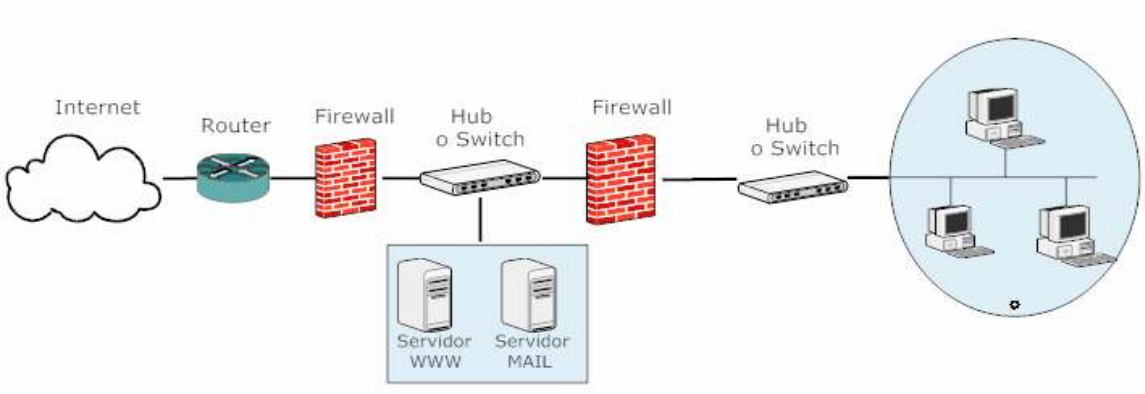


Figura 57 Esquema de firewall entre red local e Internet con zona DMZ para servidores expuestos creado con doble firewall.

Los firewalls se pueden utilizar en cualquier red. Es muy común colocarlos como protección de Internet en las empresas, aunque también en estos casos suelen tener una doble función: controlar los accesos externos hacia dentro y también los internos hacia el exterior, a esta funcionalidad se le conoce Proxy. En las empresas que poseen muchos servidores alojados, lo normal es encontrar uno o más firewalls ya sea filtrando toda la instalación o parte de ella.

En la siguiente figura, se muestra un esquema de un ISP, en el cual se coloca un firewall para proteger determinados servidores, el firewall se coloca en medio entre resto de la red y la zona de servidores protegidos.

Existen dos formas de implementar un firewall:

1) Política por defecto ACEPTAR: en principio todo lo que entra y sale por el firewall se acepta y solo se denegará lo que está previamente establecido en las políticas de seguridad.
2) Política por defecto DENEGAR: todo esta denegado, y solo se permitirá pasar por el firewall aquellos que se permitan explícitamente.

Es importante considerar que el orden en el que se colocan las reglas del firewall es determinante. Cuando se decide que se hacer con un determinado paquete, se compara con cada regla del firewall hasta que se encuentra una que le afecte luego se ejecuta el contenido de la regla (aceptar o denegar); después de ese proceso ya no se buscan otras reglas para ese paquete.

IMPORTANTE

El orden en el que se ponen las reglas de Firewall es determinante. Normalmente cuando se decide que se hace con un paquete se compara con cada regla del Firewall hasta que se encuentra una que le afecta (match), y se hace lo que dicte esta regla (aceptar o denegar); después de eso NO SE CONSIDERAN MÁS REGLAS para ese paquete. ¿Cuál es el peligro? Si se ponen reglas muy permisivas entre las primeras del Firewall, puede que las siguientes no se apliquen y no sirvan de nada.

7.4 Iptables

Iptables es un sistema de Firewall vinculado al kernel de linux que se ha extendido enormemente a partir del kernel 2.4 de este sistema operativo. Al igual que el anterior sistema ipchains, un Firewall de iptables no es como un servidor que lo iniciamos o detenemos o que se pueda caer por un error de programación. Iptables esta integrado con el kernel, es parte del sistema operativo. Realmente lo que se hace es aplicar reglas.

Para ellos se ejecuta el comando iptables, con el que añadimos, borramos, o creamos reglas. Por ello un Firewall de iptables no es sino un simple script de shell en el que se van ejecutando las reglas de Firewall. Se pueden salvar las reglas aplicadas con el comando iptables-save en un fichero.

Esquema del funcionamiento de IPTABLES

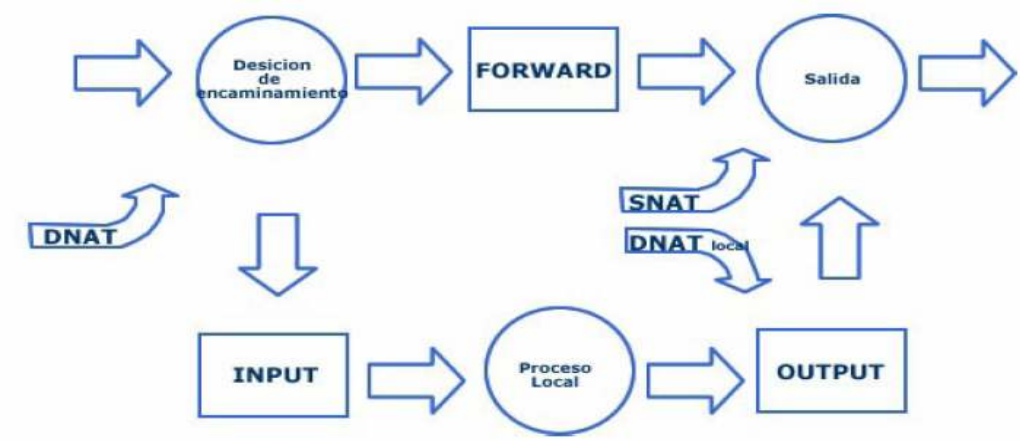


Figura 58: cuando un paquete u otra comunicación llegan al kernel con iptables se sigue este camino

Como se ve en el gráfico, se observa si el paquete esta destinado a la propia maquina o si va a otra. Para los paquetes (o datagramas, según el protocolo) que van a la propia maquina se aplican las reglas INPUT y OUTPUT, y para filtrar paquetes que van a otras redes o maquinas se aplican simplemente reglas FORWARD. INPUT, OUTPUT y FORWARD son los tres tipos de reglas de filtrado.

Tipos de reglas en IPTABLES

- **MANGLE**
- **NAT: reglas PREROUTING, POSTROUTING**
- **FILTER: reglas INPUT, OUTPUT, FORWARD.**

7.5 Netfilter6

El kernel de Linux presenta un subsistema de redes llamado Netfilter que proporciona un filtrado de paquetes con vigilancia continua, así como también servicios de NAT y de enmascaramiento IP. El manejo y flexibilidad de Netfilter es implementado a través de la interfaz de iptables. Esta herramienta de línea de comandos es similar en sintaxis a su predecesor, ipchains; sin embargo, iptables utiliza el subsistema Netfilter para mejorar la conexión de la red, inspección y procesamiento.

El diseño y construcción de firewalls es importante, especialmente si se utiliza IPv6 en redes internas con direcciones globales de IPv6. Porque a diferencia de las redes IPv4, comúnmente los host internos son protegidos automáticamente usando direcciones privadas IPv4, según la RFC 1918 (Direccionamiento local para Internet privada o direccionamiento privado y automático IP). Las reglas de filtrado para IPv6 nativo, solamente es soportado por las versiones del kernel 2.4 o superiores, si es anterior como 2.2 se puede aplicar los filtros IPv6 sobre IPv4 pro el protocolo 41[66].

7.6 Ip6tables

Ip6tables es utilizado para ejecutar, mantener e inspeccionar las tablas de los filtros de los paquetes IPv6 dentro del kernel de Linux. Se pueden definir diferentes tipos de tablas. Cada tabla contiene un número de cadenas de construcción y también puede contener cadenas definidas por el usuario. Cada cadena, es una lista de reglas las cuales coinciden con algunos paquetes. Cada especificación de la regla que se diseña debe de coincidir con un paquete IP, a dicha coincidencia se le conoce como ***target***, y sino

[66] *Ver Capítulo 1"Introducción a IPv6" tabla 2, página 27*

coincide con cierta regla, entonces salta hacia otra cadena previamente definida por el usuario dentro de la misma tabla.

TARGETS

Una regla especifica los criterios para un paquete. Si el paquete no coincide, la próxima regla será examinada, y si coincide, entonces la próxima regla es especificada por el valor del target, la cual puede haberla creado el usuario o uno de los comandos especiales: ACCEPT, *DROP*, *QUEUE*, o *RETURN*.

ACCEPT

Significa dejar pasar el paquete a través de la red. *DROP* significa descartar el paquete, *QUEUE* es dejar pasar el paquete a un espacio definido o poner en cola (depende si es soportado por el kernel). *RETURN* detiene la cadena.

TABLES

Actualmente hay dos tablas independientes (las cuales se puede hacer uso en cualquier momento, o desde la configuración previa y están presentes en cada módulo) la tabla NAT no puede implementarse todavía.

7.6.1 Configuración de ipv6 en Red Hat Enterprise

Red Hat Enterprise Linux soporta las reglas de firewall de IPv6 usando el subsistema Netfilter6 y el comando ip6tables. Para utilizar ip6tables se deben de ejecutar los siguientes pasos:

1. El primer paso en el uso de ip6tables es iniciar el servicio ip6tables. Esto se realiza a través del comando:

service ip6tables start

Las reglas guardadas para ip6tables son almacenadas en el archivo: ***/etc/sysconfig/ip6tables***

Las reglas anteriores guardadas por los scripts de inicio de ip6tables son guardadas en el archivo:
/etc/sysconfig/ip6tables.save

2. Se deben apagar los servicios iptables para utilizar exclusivamente el servicio ip6tables:
service iptables stop
chkconfig iptables off

3. Para ejecutar ip6tables por defecto cada vez que se inicia el sistema, se debe de cambiar el estado del nivel de ejecución en el servicio usando chkconfig.
chkconfig --level 345 ip6tables on

7.6.2 Elementos para construir una regla ip6table

-t, --table *tabla*
Esta opción especifica el paquete que coincide con la tabla de comandos y que será operado o manejado. Si el Kernel es configurado en modo de carga automática, entonces, se cargará el modulo para la tabla. Las tablas son las siguientes:

a. Filter:
Esta es la tabla por defecto (-t); Contiene las cadenas que se construyen **INPUT** (para paquetes que están entrando en la red), **FORWARD** (para paquetes que han comenzado a ser ruteados y transportados) y **OUTPUT** (para paquetes generados localmente).

b. Mangle:
Esta tabla es utilizada para paquetes alterados, las cadenas que se utilizan son las siguientes: **PREROUTING** (para alterar los paquetes entrantes antes del ruteo) y **OUTPUT** (para alterar localmente paquetes generados después del ruteo).

7.6.2.1 Comandos de ip6table

Estas opciones especifican la acción que se ejecuta. Solamente una de ellas puede ser declarada en la línea de comandos. Para todas las versiones de los comandos y los nombres de las opciones, solo se necesita colocar una letra del alfabeto.

-A, --append *cadena regla-especificación*
Abre una o más reglas al final de la cadena seleccionada. Cuando los nombres de la fuente y /o el destino resuelven mas de una dirección, una regla puede ser agregada para cada posible combinación de dirección.

-D, --delete *cadena regla- especificación*
-D, --delete *cadena número de regla*
Borra una o mas reglas que provienen de la cadena seleccionada. Existen dos versiones de este comando. la regla puede ser especificada como un número en la cadena (comenzando desde uno para la primera regla) o una regla que coincida.

-I, --insert
Inserta una o más reglas en la cadena seleccionada según el numero de la regla, Si el número es uno, la regla o reglas son insertadas en la cabecera de la cadena. También por defecto si el número de la regla no es especificado.

-R, --replace cambia el número de la regla
Reemplaza una regla, por otra. Si la fuente y / o el destino posee múltiples direcciones, el comando fallará. Las reglas deben de comenzar desde 1.

-L, --list [*cadena*]
Muestra en una lista todas las reglas que fueron seleccionadas en una cadena. Si la cadena no se encuentra seleccionada busca a la tabla especificada (filter por defecto), o las reglas mangle, que se obtienen de la lista de ip6tables -t mangle -n –L se debe de considerar que frecuentemente es usada con la opción –n, en orden para evitar los congestionamientos de DNS.

-F, --flush [*cadena*]
Es equivalente a borrar todas las reglas una por una.

-Z, --zero [*cadena*]
El paquete Zero y los contadores byte en todas las cadenas. Esto es legal para especificar la **-L, --list** (list) opción, para ver las cuentas inmediatamente cuando estas son declaradas.

-N, --new-cadena *cadena*
Crea un Nuevo usuario definido.

-X, --delete-cadena [*cadena*]
Borra las opciones de los usuarios definidos en la cadena especificada. No debe de referirse a la cadena; si con anterioridad se elimina o se reemplaza las reglas referidas, antes de que la cadena pueda ser borrada.

-P, --policy *cadena target*
Coloca una política de la cadena, Solamente construye (no utiliza usuarios definidos).

-E, --rename-cadena *old-cadena new-cadena*
Renombra a los usuarios especificados dentro de una cadena.

-H, --help
Brinda una breve descripción de la sintaxis de los comandos.

7.6.2.2 Parámetros

Los siguientes parámetros se utilizan para hacer referencia a una regla (como usarla, agregar, borrar, inserter, reemplazar).

-p, --protocol [!] *protocol*

El protocolo de una regla o de un paquete a verificar. EL protocolo específico puede ser *tcp*, *udp*, *icmpv6*, o todos, o se puede colocar el valor numérico. El nombre de un protocolo desde la ruta /etc/protocols tambien es permitido. El símbolo **"!"**, Indica un argumento antes del protocolo e invierte la lectura y el chequeo. El numero cero es equivalente a todo.

-s, --source [!] *address*[/*mask*]

Específica el host fuente. *Address* puede representar cada nombre del host, una dirección de red IPv6 (Con /mascara), o una dirección plana (el nombre de la red no la soporta). Aunque, una mascara de 64 es equivalente a *ffff:ffff:ffff:ffff:0000:0000:0000:0000*. Un argumento "!" antes de la especificación de la dirección invierte la sensibilidad de la dirección. La bandera **--src** es un alias para esta opcion.

-d, --destination [!] *address*[/*mask*]

Las especificaciones del destino **-s** (fuente) de bandera es para detallar la descripción de la sintaxis. La bandera **--dst** es un alias para esta opción.

-j, --jump *target*

Especifica las reglas empleadas para el target, que se usan cuando el paquete coincide.

-i, --in-interface [!] *name*

Nombre de una interfaz que recibe un paquete (se aplica solamente para paquetes que tienen cadenas de tipo input, forward y prerouting). Cuando se coloca el argumento "!" y es usado antes del nombre de la interfaz, se invierte la sensibilidad. Si el nombre de la interfaz finaliza con "+", entonces cualquier interfaz que comience con este nombre deberá de coincidir.

-o, --out-interface [!] *name*

Nombre de una interfaz; se aplica para los paquetes que se envían (el paquete debe de estar marcado por las sintaxis **FORWARD** y **OUTPUT** c). Cuando el argumento "!" es usado antes del nombre de la interfaz, la sensibilidad es invertida. Si la el nombre de la interfaz finaliza con "+", entonces cualquier interfaz podrá comenzar con este nombre que coincidirá, Si esta opcion es omitida, cualquier nombre de la interfaz coincidirá

-c, --set-counters PKTS BYTES

Habilita al administrador para que inicialice el paquete y las cuentas de la regla (durante las operaciones de **INSERT, APPEND, REPLACE**).

7.6.2.3 Match Extensions

Ip6tables puede ser usado para extensiones de paquetes de los módulos. Estos son cargados de dos formas diferentes: cuando se especifica **-p** o **--protocol** , o con **-m** or **--match** , seguido por el nombre del módulo que coincida; después de esto, varias líneas de comando extra comienzan a ser habilitadas , dependiendo del modulo especifico. Se puede especificar varios módulos en una sola línea, y se puede utilizar **-h** o **--help** después que los módulos se hayan especificado para recibir ayuda. Los siguientes son incluidos dentro de la base del paquete, y muchos de ellos pueden preceder por el símbolo **! El cual invierte la sensibilidad del manejo.**

a. tcp

Esta extensión es cargada si el protocolo TCP es especificado. Las siguientes son las opciones de este comando:

--source-port [!] *port*[:*port*]

Puerto fuente o puerto de una dirección específica, también puede recibir el nombre o número del puerto, también representa un rango de direcciones, se utiliza el formato port: port. Si el primer puerto es omitido se asume que posee el valor de “0”, y si el último se omite, entonces tiene el valor de “65535”. La bandera **–-sport** es un alias conveniente para ésta opción.

--destination-port [!] *port*[:*port*]

Puerto destino o un rango de puerto. La bandera **–dport** es conveniente colocarle un alias a esta opción.

--tcp-flags [!] *mask comp*

Coinciden cuando las banderas TCP son especificadas. El primer argumento pertenece a las banderas que deben de ser examinadas. Se escriben en una lista, separadas por una coma.

b. Udp

Estas extensiones son cargadas si el protocolo UDP es previamente especificado. Provee las siguientes opciones:

--source-port [!] *port*[:*port*]

Puerto de la fuente o rango específico para el puerto.

--destination-port [!] *port*[:*port*]

Puerto destino o un rango específico para el puerto.

c. ipv6-icmp

Esta extensión es ejecutada si el –protocolo IPv6- icmp o el protocolo ICMv6 es especificado. Esto provee las siguientes opciones:

--icmpv6-type [!] *typename*

Especifica el tipo de de ICMP, el cual puede ser ipv6-ICMP de tipo numérico, o un icmp-ipv6 de tipo nombre y se presenta por el siguiente comando comando

ip6tables -p ipv6-icmp -h

d mac

--mac-source [!] *address*

Debe de coincidir con la dirección MAC del origen, y debe de tener el siguiente formato: XX:XX:XX:XX:XX:XX. Note que esta solamente hace por sensiblidad que poseen los

paquetes que entran en dispositivos Ethernet y que entran en los procesos de cadena: **PREROUTING**, **FORWARD** o **INPUT**.

e. multiport

Este modulo debe de coincidir con los puertos del destino o los del origin. Pueden especificarse hasta 15 puertos. Esto puede ser posible, solamente utilizado con la instrucción **-p tcp** o **-p udp**.

--source-ports *port*[,*port*[,*port*...]]

Coincide con el Puerto del origen, si existe en los puertos especificados. La bandera **--sports** es un alias conveniente para esta opción.

--destination-ports *port*[,*port*[,*port*...]]

Si coincide con el puerto destino y éste ya esta previamente definido en el listado. La bandera **--dports** es un alias conveniente para esta opción.

--ports *port*[,*port*[,*port*...]]

Coincide si ambos puertos tanto el origen como el destino son iguales o cada distinto del otro.

e. mark

Este modulo coincide con la marca de netfilter que marca el campo especificado con un paquete.

f. owner

Este modulo relaciona varias características del paquete creador, para localizar los paquetes que han sido generados. Esto es solamente valido en la cadena **OUTPUT**, y algunos paquetes (como respuestas ping empleando ICMP) no pueden tener ningún propietario, y entonces nunca van a coincidir Este comando es considerado como experimental.

--uid-owner *userid*

Coincide si el paquete y fue creado por un proceso con la efectividad que el usuario asignó.

--gid-owner *groupid*

Coincide si el paquete fue creado; por un proceso que fué generado por la efectividad de una identificación del grupo.

--pid-owner *processed*

Coincide si el paquete fué creado por un proceso a traves de un identificación del proceso

--sid-owner *sessionid*

Coincide si el paquete fue creado por un proceso a través de la sesión del grupo.

7.7.2.4 Instrucciones

- Para reglas cortas

ip6tables -L

- Reglas extendidas

ip6tables -n -v --line-numbers -L

- Lista específica de filtros

p6tables -n -v --line-numbers -L INPUT

- Inserta una regla en la entrada del filtro con diferentes opciones:

ip6tables --table filter --append INPUT -j LOG --log-prefix "INPUT:"

¬ --log-level 7

- Eliminar una regla en la entrada del filtro

ip6tables --table filter --append INPUT -j DROP

- Eliminar una regla según el número

ip6tables --table filter --delete INPUT 1

- Aceptar un paquete entrante de icmpv6 a través de un túnel

ip6tables -A INPUT -i sit+ -p icmpv6 -j ACCEPT

- Permitir un paquete saliente de icmpv6 a través de un túnel
ip6tables -A OUTPUT -o sit+ -p icmpv6 -j ACCEPT

- Ahora el Nuevo Kernel, permite especificar el tipo de mensaje icmpv6
i***p6tables -A INPUT -p icmpv6 --icmpv6-type echo-request -j ACCEPT***

Protección para peticiones TCP

Para la implementación de la seguridad, se recomienda insertar una regla en forma de bloques para las peticiones entrantes de TCP.

- Para bloquear peticiones TCP entrantes a un determinado host
ip6tables -I INPUT -i sit+ -p tcp --syn -j DROP

- Para bloquear peticiones TCP entrantes para host dependientes de un router.
ip6tables -I FORWARD -i sit+ -p tcp --syn -j DROP

Capítulo

8

Calidad de servicio QoS.

Temas:

- Calidad de servicio (QoS).
- Beneficios de QoS.
- Arquitectura Básica de las QoS.
- Calidad de Servicio en IPv6.
- Aplicando QoS en hosts y routers

8. Calidad de servicio (QoS)

En términos generales, puede definirse la Calidad del Servicio (QoS) como la capacidad que tiene un sistema de asegurar, con un grado de fiabilidad preestablecido, que se cumplan los requisitos de tráfico, en términos de perfil y ancho de banda, para un flujo de información dado. Más específicamente, para el caso de proveedores de red, se establece en el RFC 2475 (An Architecture for Differentiated Services).

Donde un Servicio define algunas características significativas de la transmisión de un paquete en una dirección, a través de un conjunto de una o más rutas dentro de la red. Estas características pueden especificarse en términos de caudal (throughput), retardo (delay), variación de demora (jitter) y/o pérdidas, o también en términos de alguna prioridad relativa de acceso a los recursos de la red.

En aplicaciones demandantes tales como sesiones de Chat de voz y video en tiempo real solo se toleran retardos de fracciones de segundos para satisfacer los requerimientos humanos. Es importante conocer las aplicaciones que utilizan la red. Una vez se conocen todos estos elementos se decide que tipo de Q oS se implementará. Por ejemplo: listas de acceso, que pueden ser usuradas para identificar a los requerimientos prioritarios para accesar a los recursos de la red.

8.1 Beneficios de QoS

Al implementar calidad de servicio se atienen los siguientes beneficios, no importa si la corporación o empresa sea grande, mediana o pequeña:

- **Control de los recursos:** se debe tener control sobre los siguientes recursos: ancho de banda, equipo, tamaño de la red, facilidades.

- **Eficiencia al utilizar los recursos de la red**: si se cuenta con un buen análisis sobre el manejo de la red y las herramientas necesarias para su desarrollo, se puede obtener con facilidad los servicios de red e identificar el tráfico más importante de la empresa.

- **Coexistencia de las tareas y aplicaciones críticas:** cuando se emplean QoS en las redes WAN, se obtiene eficiencia en el manejo de las tareas criticas de las aplicaciones, ancho de banda y mínimo retardo requerido por un tiempo sensible y requerido en el uso de multimedia y aplicaciones con voz.

- **Incremento de servicios en la red:** las características de QoS incrementan y hacen que las redes sean predecibles para ofrecer los siguientes servicios :
    - Ancho de banda dedicado
    - Manejo de la congestión
    - Priorización del tráfico

8.2 Arquitecturas de QoS

Se denominan arquitecturas de QoS a aquellas que lo ofrecen. Se pueden distinguir los siguientes modelos:

a) Arquitectura de Servicios Integrados

En este modelo, la aplicación enmarca su solicitud de servicio dentro de un Protocolo de Reserva de Recursos (RSVP), y entonces pasa su solicitud a la red. En este tipo de arquitectura, cada estación o router en el camino de los datos debe manejar las peticiones de reservas y luego debe asociar una espera al flujo requiriendo la reserva. Por medio de este método, la QoS puede ser garantizada, pero los “Servicios Integrados” son complejos de implementar y pueden generar mucho tráfico de señalización. Este exceso de tráfico de señalización lleva a una pobre escalabilidad para el modelo de Servicios Integrados

b) Arquitectura de Servicios Diferenciados (DiffServ): se basa en la separación de los conceptos básicos de operación de los encaminadores de reenvío (forwarding) y control (encaminamiento). En el reenvío se realiza un tratamiento diferenciado de los datagramas. La interacción entre la red y la aplicación toma la forma de una solicitud de servicio sin negociación previa, en la que la aplicación solicita un servicio marcando cada paquete con un código que indica el servicio deseado.

8.3 Calidad de Servicio en IPv6

El protocolo Ipv6 fue diseñado para mejorar las cualidades de Calidad de Servicio. Se crean dos nuevos campos y se mejora el rendimiento general del protocolo. La mejora del rendimiento se basa en que los paquetes pueden ser tratados de manera eficiente por los routers. IPv6 soporta la Calidad de Servicio a través de dos campos nuevos: etiqueta de flujo y clase de tráfico. Al introducir el campo de Etiqueta de Flujo, IPv6 clasifica los paquetes de acuerdo a su destino y a su servicio.

8.3.1. Campo Clase de Tráfico (8 bits)

Se denomina prioridad o simplemente clase. Este campo debe ser usado por los nodos que originan el tráfico o por los routers que se utilizan para identificar y distinguir entre diferentes clases y prioridades de los paquetes IPv6, los cuales deben recibir un tratamiento particular, en cada nodo. Este campo fue diseñado para soportar Servicios Diferenciados (DiffServ).

El campo Clase de Tráfico (TC) de 8 bits es usado por los nodos y routers para identificar y distinguir los paquetes enviados con clases diferentes o prioritarias. Los valores que pueden tomar estos bits dependen de los nodos que envían los paquetes, el valor por omisión en este campo es de cero.

8.3.2. Campo Etiqueta de Flujo (20 bits)

Este campo puede ser usado en el origen para etiquetar los paquetes, en un mismo destino, que necesiten de un tratamiento especial por los routers IPv6, tales como servicios de QoS no habituales o “real time”. Esta herramienta es todavía experimental y sujeta a cambios hasta que se encuentren claros los requerimientos de soporte de flujo en Internet. Este campo fue diseñado para soportar Servicios Integrados (IntServ).

Los 20 bits del campo etiqueta de flujo pueden ser usados por la fuente para etiquetar la secuencia de paquetes para los cuales se requiere un trato especial por los routers IPv6, por ejemplo un servicio de calidad diferente de la normal o algún servicio de tiempo real. La naturaleza de este trato especial debe ser cubierta por los routers por medio de un protocolo de control (RSVP[67]-RFC 2205), o por la misma información de la etiqueta de los paquetes, por ejemplo, la opción Salto-a-Salto.

[67] *Protocolo (RSVP), Reserva Recurso, especificado en la RFC 2205*

Capítulo

9

Implementación de IPv6 con dispositivos CISCO

Temas:

- **Conectividad básica en IPv6.**
- **Verificando la conectividad básica en Ipv6.**
- **Calidad de servicio para Ipv6**
- **VPN con Ipsec de IPv6**

9. Implementación del protocolo Ipv6 con dispositivos CISCO

Cisco ha creado una serie de Sistemas Operativos para sus plataformas los cuales dan soporte a características importantes de IPv6. A continuación se muestra en la siguiente tabla las herramientas que dan soporte según el tipo de IOS que posea el dispositivo.

Características	Cisco IOS Requeridos
Características básicas de ipv6 para Cisco IOS.	12.2(2)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T, 12.4(2)T
Formato de direcciones Ipv6.	12.2(2)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T, 12.4, 12.4(2)T
Direcciones Unicast Ipv6.	12.2(2)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T, 12.4, 12.4(2)T
DNS for IPv6	12.2(2)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T, 12.4, 12.4(2)T
Map host names to IPv6 addresses	12.2(2)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T, 12.4, 12.4(2)T
IPv6 path MTU discovery	12.2(2)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T, 12.4, 12.4(2)T
ICMPv6	12.2(2)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T, 12.4, 12.4(2)T
IPv6 neighbor discovery	12.2(2)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T, 12.4, 12.4(2)T
IPv6 stateless autoconfiguration	12.2(2)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3,12.3(2)T, 12.4, 12.4(2)T

ATM PVC and Frame Relay PVC3	12.2(2)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T, 12.4, 12.4(2)T
FDDI	12.2(2)T, 12.2(14)S, 12.3, 12.3(2)T, 12.4, 12.4(2)T
PPP service over packet over SONET, ISDN, and serial (synchronous and asynchronous) interfaces	12.2(2)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T, 12.4, 12.4(2)T
Ethernet, Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet	12.2(2)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T, 12.4, 12.4(2)T
Dual IPv4 and IPv6 protocol stacks	12.2(2)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T, 12.4, 12.4(2)T
Configuring IPv6 addressing and enabling IPv6 routing4	12.2(2)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T, 12.4, 12.4(2)T
Cisco High-Level Data Link Control (HDLC)	12.2(2)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T, 12.4, 12.4(2)T
ICMPv6 redirect	12.2(4)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T, 12.4, 12.4(2)T
IPv6 neighbor discovery duplicate address detection	12.2(4)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T, 12.4, 12.4(2)T
DNS lookups over an IPv6 transport	12.2(8)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T, 12.4, 12.4(2)T
ICMPv6 rate limiting	12.2(8)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T, 12.4, 12.4(2)T
CISCO-IP-MIB support	12.0(22)S, 12.2(14)S, 12.2(15)T, 12.3,

	12.3(2)T, 12.4, 12.4(2)T
CISCO-IP-FORWARDING-MIB support	12.0(22)S, 12.2(14)S, 12.2(15)T, 12.3, 12.3(2)T, 12.4, 12.4(2)T
Dynamic packet transporte (DPT)	12.0(23)S
Unicast Reverse Path Forwarding (Unicast RPF) strict mode	12.2(13)T, 12.2(14)S, 12.3, 12.3(2)T, 12.4, 12.4(2)T
IPv6 address types: Anycast	12.3(4)T, 12.2(25)S, 12.4, 12.4(2)T
DHCP for IPv6 prefix delegation	12.3(4)T, 12.4, 12.4(2)T
Stateless DHCP for IPv6	12.3(4)T, 12.4, 12.4(2)T
Remote bridged encapsulation (RBE)	12.3(4)T, 12.4, 12.4(2)T
NetFlow for IPv6	12.3(4)T, 12.4, 12.4(2)T
Unicast Reverse Path Forwarding (Unicast RPF)	12.2(25)S
DHCP for IPv6 Relay Agent	12.3(11)T, 12.4, 12.4(2)T
IP6.ARPA support was added.	12.3(11)T
IPv6 default router preferences	12.4(2)T
Syslog for IPv6	12.4(2)T
HSRP for IPv6	12.4(2)T

Las versiones de IOS 12.2 (T) y 12.2 (14) son las que soportan las principales características de Ipv6. Por lo tanto se recomienda que el equipo que se utilice deba de poseer dichas versiones. En la siguiente tabla se muestra las plataformas de Router Cisco que soportan estas versiones de IOS.

PLATAFORMA	12.2 T Releases	
	12.2 (2) T	12.2 (14) S
Cisco 800 Series	Si	Si
Cisco 1400 Series	Si	Si
Cisco 1600 Series	Si	Si
Cisco 1700 Series	Si	Si
Cisco 2500 Series	No	Si
Cisco 2600 Series	Si	Si
Cisco 2800 Series	Si	Si
Cisco 3600 Series	Si	Si
Cisco 4000 Series	Si	No
Cisco 7100 Series	Si	Si
Cisco 7200 Series	Si	Si
Cisco 7500 Series	Si	No
Cisco 1200 Series	No	No

9.1 Conectividad básica en IPv6.

Existen muchos métodos para configurar una dirección IPv6 para cada interfaz:

- El método simple, consiste en habilitar la propiedad de autoconfiguración en las interfaces. Esta configuración se basa en los prefijos recibidos de los mensajes de advertencia que generan los Routers automáticamente. Una dirección de enlace local basada en Identificador de la interfase EUI-64[68], es automáticamente generada por la interfaz cuando la autoconfiguracion es habilitada. Para habilitarla solo se ingresa el siguiente comando:
  Hostname (config-if)# **ipv6 address autoconfig.**

- Si solamente se necesita configurar una dirección de enlace local, en una interfaz y no registrar otra dirección IPv6. Se Ingresa el siguiente comando para especificar una dirección de enlace local.
  hostname(config-if)# **ipv6 address** ipv6-address **link-local**

- El siguiente comando se utiliza para habilitar el protocolo IPv6 y automáticamente genera la dirección del enlace local, utilizando EUI-64 el cual se basa en la dirección MAC:
  hostname(config-if)# **ipv6 enable**

Cuando una interfaz en un dispositivo de red de Cisco es configurada con ambas direcciones IPv4 e IPv6 dicha interfaz envía tráfico de paquetes IPv4 e IPv6, además puede enviar y recibir datos desde redes IPv6 e IPv4.
Para configurar una interfaz de un dispositivo de red Cisco de tal manera que soporte la pila de protocolo IPv4 e IPV6 se debe utilizar la siguiente configuración desde el modo de configuración global.

[68] *El Institute of Electrical and Electronic Engineers (IEEE) define la dirección EUI-64 de 64 bits. Las direcciones EUI-64 se asignan a un adaptador de red o se derivan de las direcciones IEEE 802.*

Configuración básica

Paso	Comando	Propósito
1	**enable** **Ejemplo:** Router> enable	Habilita el modo Exec privilegiado.
2	**configure terminal** **Ejemplo:** Router# configure terminal	Accede al modo de configuración global.
3	**interface** type number **Ejemplo:** Router(config)# interface ethernet 0/0	Especifica un tipo y un número de interfaz.
4	**ipv6 address** ipv6-prefix/prefix-length **eui-64** **Ejemplo:** Router(config-if)# ipv6 address 2001:0DB8:0:1::/64 eui-64 **ipv6 address** ipv6-address **link-local** **Ejemplo:** Router(config-if)# ipv6 address FE80::260:3EFF:FE11:6770 link-local **ipv6 enable** Router(config-if)# ipv6 enable	Especifica la dirección, el tipo de interfaz y entra al modo de configuración de interfaz. Especifica una dirección ipv6 asignada ala interfaz y habilita el proceso de reconocimiento del protocolo en esa interfaz Habilita el protocolo en la interfaz
5	**exit** **Ejemplo:** Router(config-if)# exit	Ejecuta la salida del modo de configuración en una interfaz y regresa al modo de configuración global.

6	**ipv6 unicast-routing** **Ejemplo:** Router(config)# ipv6 unicast-routing	Habilita el envío de datagramas unicast IPv6

9.2 Verificando la conectividad básica en Ipv6.

Para verificar y comprobar la conexión entre los dispositivos de una red, se puede auxiliar de los siguientes comandos:

a) Show ipv6 interface

Sintaxis:
Show ipv6 interface [**brief**] [[interface-type interface-number] [**prefiex**]]

El comando show ipv6 interface es utilizado para verificar que las direcciones Ipv6 se encuentran configuradas correctamente, en una interfaz del router. También se puede visualizar la información referente al estado de los Mensajes de redirección de vecinos (Ipv6 neighbors Message, Redirect Messages) ICMPV6, utilizados para proveer información del siguiente salto para una ruta hacia un destino. Además de los mensajes de descubrimiento de vecinos y de la configuración sin estado.
Cuando se ejecuta el comando Show IPv6 interface, se muestra la siguiente información:

- El nombre y estado de la interfaz.
- El enlace local y global de la dirección unicast.
- El grupo multicast de la interfaz.
- Redireccionamiento ICMP y los mensajes de errores.
- Descubrimiento de vecinos.

b) **Show Ipv6 Neighbors**

Sintaxis:
Show Ipv6 neighbors [interface-type interface-number |ipv6-address]

El comando **Show Ipv6 neighbors** se utiliza para desplegar en pantalla la información del caché de descubrimiento de vecinos. El signo (-), indica que es una entrada estática.

c) Show ipv6 Route.

Sintaxis
Show ipv6 route [ipv6- address | ipv6- prefix / prefix-length | protocol]

Este comando muestra el contenido de la tabla de enrutamiento para una dirección Ipv6 específica. El resultado de la ejecución de este comando (show ipv6 route) es similar a la que se emplea para el protocolo ipv4 (show route). Muestra la información siguiente:

- La ruta que ha seguido el protocolo
- El prefijo Ipv6 de la red remota
- La distancia administrativa y la métrica de la ruta.
- La dirección del próximo salto del router.
- La interfaz donde se dará el próximo salto.

d) Show Ipv6 Traffic

Este comando muestra estadísticas acerca del tráfico Ipv6. Se utiliza el comando para desplegar información sobre los contadores de límites de tasa de ICMP.

e) Show running-config

El comando show running-config es utilizado para verificar que el procesamiento de paquetes Ipv6 se encuentre habilitado globalmente en el router y sobre las interfaces aplicables, además de la configuración de una dirección IPv6 sobre una interfaz.

9.3 Calidad de servicio para Ipv6

Es una característica de una red de telecomunicaciones que permite garantizar al cliente una calidad pactada por cada servicio contratado[69].

9.3.1 Prerrequisitos para implementar QoS para Ipv6.

La siguiente tabla muestra el sistema el operativo de Cisco requerido para soportar las diferentes características de QoS.

Característica	Mínimo Cisco Requerido
Calidad de servicio para Ipv6	12.2(13)T, 12.3, 12.3(2)T, 12.0(28)S1, 12.4, 12.4(2)T
Línea de Comando Modular para QoS (MQC)	12.2(13)T, 12.3, 12.3(2)T, 12.4, 12.4(2)T
Manipular el tráfico MQC	12.2(13)T, 12.3, 12.3(2)T, 12.0(28)S1, 12.4, 12.4(2)T
Establecer políticas de Tráfico MQC	12.2(13)T, 12.3, 12.3(2)T, 12.0(28)S1, 12.4, 12.4(2)T
Marcar paquetes MQC	12.2(13)T, 12.3, 12.3(2)T, 12.0(28)S1, 12.4, 12.4(2)T
Encolamiento	12.2(13)T, 12.3, 12.3(2)T, 12.0(28)S1, 12.4, 12.4(2)T
MQC WRED	12.2(13)T, 12.3, 12.3(2)T, 12.0(28)S1, 12.4, 12.4(2)T

[69] *Ver capítulo 8 "Calidad de servicio QoS"*

9.3.2 Estrategia para implementar calidad de servicio sobre Ipv6

Las características de calidad de servicio son soportadas por ambientes Ipv6 incluyen la clasificación de paquetes, encolamiento, prevención de la congestión, marcación de paquetes y políticas de enrutamiento para paquetes Ipv6. Todas las características de QoS disponible para ambientes Ipv6 son administradas desde la interfaz de línea de comando (CLI) para QoS.

Definida en el sistema Operativo de Cisco; la cual permite crear clases de tráfico, políticas de tráfico (policy maps) y agregar estas mismas a interfaces específicas de los enrutadores. Antes de implementar calidad de servicio en redes Ipv6, se deben determinar ciertos aspectos importantes, los cuales son:

- Conocer las aplicaciones que necesitan QoS en la red.
- Entender las características de las aplicaciones para así, poder tomar decisión como se va a implementar la calidad de servicio.
- Conocer la topología de la red para así poder conocer como son afectados los tañaos de los encabezados de capa de enlace.
- Crear clases basadas en los criterios que se establecieron para la red. En particular, si en la misma red se está llevando tráfico Ipv4 e IPv6, se puede decidir el trato que se le dará a cada tipo de tráfico.

Si se configurar el tráfico Ipv4 e ipv6 se deben utilizar los comandos:

1. **Match precedent**
2. **Match dscp**
3. **Set precedent**
4. **Set dscp**

Si se quiere configurar por separado:

A. Se debe agregar un criterio de comparación como:

- **Match protocolo id**
- **March protocolo ipv6**

B. Crear una política para marcar cada clase.
C. Trabajar desde las fronteras hacia el núcleo aplicando políticas de QoS.
D. Construir las políticas para tratar el tráfico.
E. Aplicar la política.

9.3.3 Clasificación de paquetes en IPv6

La clasificación de paquetes se encuentra disponible en los procesos de conmutación. Se basa en una precedencia IPv6 (id Precedente), punto de control de servicios diferenciados (DSCP) y otros valores específicos del protocolo IPv6 que pueden ser especificados en listas de acceso IPv6. Una vez que se determina las aplicaciones se necesita de QoS, se pueden crear clases basadas en las características de las aplicaciones. Existe una cantidad muy variada de criterios de comparación para clasificar el tráfico y se pueden combinar varios criterios de comparación para aislar y diferenciar el tráfico.

Políticas y marcación de paquetes en redes IPv6.

Se pueden crear políticas para marcar el tráfico con valores de prioridad apropiados, utilizando precedencia y DSCP. La marcación de paquetes IPv6 permite establecer un valor de precedencia o de DSCP al tráfico para facilitar su administración.
El tráfico es marcado en el momento que entra al enrutador por una interfaz específica y ésta marca se utiliza para darle un trato específico al tráfico al salir del enrutador. Siempre se debe marcar el tráfico lo más cerca posible del origen.

Para marcar los paquetes se utilizan los comandos siguientes, los cuales han sido modificados para manipular el tráfico IPv4 e IPv6.

- **set dscp**
- **set precedent**

9.3.4 Implementación calidad del servicio para IPv6

Restricciones para clasificar el tráfico en redes IPv6

A excepción de las modificaciones del comando **match dscp**, **match precedent** y la adición del comando **match Acces-group name** específico para IPv6, la funcionalidad de todos los comandos match es la misma que para IPv4.

El comando **match access-group** que se utiliza para comparar listas de control de acceso numeradas, no es soportado.
Los comandos **set cos** y **match cos** para interfaces tipo 802.1Q[70] (dot1Q) son soportados solamente para los paquetes conmutados por CEF (Cisco Express Forwarding). Los paquetes de conmutación de procesos, como los generados por el enrutador, no son marcados cuando estas opciones son utilizadas.

a) Especificando un criterio de Marcación para los paquetes IPv6

Para marcar los paquetes se utiliza al comando **set precedente**, este valor será utilizado posteriormente para comparar los paquetes y clasificar el trafico de la red. Los comandos utilizados para éste propósito pueden ser **set procedente** o **set dscp**. Estos comandos han sido modificados para poder marcar tráfico IPv6.

b) Usando Criterios de coincidencia para manejar el flujo de tráfico de IPv6

Una vez que se definen las clases de tráfico y se establece las políticas, se puede usar los comandos match, para verificar el tráfico que posee coincidencia con las políticas de seguridad previamente establecidas.

d) Verificando los criterios de los paquetes marcados

Se utiliza el comando de **show policy.** La información interesante desde la salida de este comando es la diferencia del numero de los paquetes totales versus el numero de paquetes marcados.

[70] *El protocolo IEEE 802.1Q, fue un proyecto del grupo de trabajo 802 de la IEEE para desarrollar un mecanismo que permite a múltiples redes con puentes para compartir transparentemente el mismo medio físico sin problemas de interferencia entre las redes que comparten el medio; conocido como dot1Q.*

Los siguientes pasos de configuración son utilizados para marcar los paquetes IPv6.

1. enable
2. configure terminal
3. policy map *policy-map-name*
4. class {*class-name* | **class-default**}
5. set precedence <0-7[class-default]>
6. set dscp <0-63 [valor de servicio diferenciado]>

Estos comandos se aplican para paquetes Ipv4 e Ipv6. La acción solamente ocurre en el paquete que coincide con el criterio especificado por medio del comando **class name**, que se visualiza en el paso 4.

A continuación se presenta una tabla con los pasos en forma detallada:

Paso	Comando	Propósito
1	**Enable** **Ejemplo**: Router> enable	Habilita el modo EXEC privilegiado.
2	configure Terminal **Ejemplo:** Router# configure terminal	Acceso al modo de configuración global,
3	**policy map** policy-map-name **Ejemplo:** Router(config)# policy-map policy1	Crea una política de mapa (Policy Map), con un nombre específico y entra en el modo de configuración.
4	**class** {class-name | **class-default**} **Ejemplo:** Router(config-pmap)# class class-default	Especifica el tratamiento para el tráfico de una clase específica y entra al modo de configuración de la clase para la política de mapa.
5	**Set precedence** <0-7> [valor de precedencia]> O también: **Set dscp** <0-63>[valor de servicio diferenciado]>	Establece el valor de precedencia. Entre mas alto sea el valor, más prioritario es el tráfico.

Para verificar la configuración se puede utilizar los siguientes comandos:

- **Show policy-map interface**
- **Show policy-map**
- **Show cef interface**
- **Show ipv6 cef**
- **Show ipv6 interface neighbors**
- **Show interface statistics**

Una vez se hayan definido las clases de tráfico y las políticas, se puede utilizar el comando **match** para comprar el trafico con las políticas establecidas anteriormente. Se pueden utilizar varios criterios de comparación y dependiendo del tipo de clase se puede especificar si se desean comparar todas las clases o algunas de ellas.

Los siguientes pasos se utilizan para realizar criterios de comparación sobre el tráfico Ipv6:

1. **enable**
2. **configure terminal**
3. **class-map** {class-name | class-default}
4. **match precedence** precedence-value [precedence-value precedence-value]

O también:

match access-group name ipv6-access-group

O también:

match [**ip**] **dscp** dscp-value [dscp-value dscp-value dscp-value dscp-value dscp-value dscp-value dscp-value]

A continuación se muestran estos pasos en forma detallada:

Paso	Comando	Propósito
1	**Enable** **Ejemplo**: Router> enable	Habilita el modo EXEC privilegiado.
2	configure Terminal **Ejemplo:** Router# configure terminal	Acceso al modo de configuración global.
3	**class** {class-name | **class-default**} **Ejemplo:** Router(config-pmap-c)# class class-clsl	Crea una clase específica y entra al modo de configuración de la clase de mapa (class-map)
4	**match precedence** precedence-value [precedence-value precedence-value] O también: **match access-group name** ipv6-access-group O también **match [ip] dscp** dscp-value [dscp-value dscp-value dscp-value dscp-value dscp-value dscp-value dscp-value] **Ejemplo:** Router(config-pmap-c)# match precedence 5 Router(config-pmap-c)# match access-group name ipv6acl Router(config-pmap-c)# match ip dscp 15	Compara el valor de precedencia, el cual es aplicado al tráfico IPv4 e IPv6. Especifica el nombre de una lista de control de acceso IPv6 contra la cual se van a comparar los paquetes para determinar si pertenece a la clase de tráfico. Identifica un valor IP DSCP de comparación específica.

9.4 VPN con IPsec

La funcionalidad del manejo del protocolo IPsec con el IOS de dispositivos CISCO, proveen datos encriptados en el nivel de empaquetamiento de IP. Este manejo ofrece estándares robustos basados en soluciones de seguridad. IPsec provee autenticación en los datos y confidencialidad en los servicios. IPsec es un componente obligatorio[71] de las especificaciones de IPv6.

Requisitos para implementar IpSec para la seguridad de IPv6

Característica	Mínimo Cisco IOS Requerido
IPv6 para el uso del protocolo OSPF V3 (Primero el camino abierto más corto) por medio de IpSec	12.3(4)T, 12.4, 12.4(2)T
IPv6 IPSec VPN	12.4(4)T

Pasos para implementar IPsec[72]

1. enable
2. configure terminal
3. crypto isakmp policy *priority*
4. authentication {**rsa-sig** | **rsa-encr** | **pre-share**}
5. hash {**sha** | **md5**}
6. group {**1** | **2** | **5**}
7. encryption {**des** | **3des** | **aes** | **aes 192** | **aes 256**}
8. lifetime *seconds*
9. exit
10. crypto isakmp key *password-type keystring* {**address** *peer-address* [*mask*] | **ipv6** {*ipv6-address*|*ipv6-prefix*} | **hostname** *hostname*} [**no-xauth**]
11. crypto keyring *keyring-name* [**vrf** *fvrf-name*]

[71] *Ver capítulo 5 "**Infraestructura de IPSec** "*
[72] *Ver capítulo 5 Infraestructura de IPSec, sección 56 "Internet Key Exchange (IKE)"*

12. pre-shared-key {**address** *address* [*mask*] | **hostname** *hostname* | **ipv6** {*ipv6-address* | *ipv6-prefix*}}
key *key*

A continuación se presentan los pasos en forma detallada:

Paso	Comando	Propósito
1	**Enable** **Ejemplo**: Router> enable	Habilita el modo EXEC privilegiado.
2	configure Terminal **Ejemplo:** Router# configure terminal	Acceso al modo de configuración global.
3	**crypto isakmp policy** priority **Ejemplo:** Router(config)# crypto isakmp policy 15	Define una política IKE, y entra al modo de configuración de la política ISAKMP. El numero 1 de la política, indica que la prioridad de la política es alta. Si el valor es menor, entonces la prioridad será mayor.
4	**authentication** {**rsa-sig** | **rsa-encr** | **pre-share**} **Ejemplo:** Router(config-isakmp-policy)# authentication pre-share	Especifica el método de autenticación con una política IKE. El comando **rsa-sig** y **rsa-encr** no son soportadas en Ipv6.
5	**hash** {**sha** | **md5**} **Ejemplo:** Router(config-isakmp-policy)# hash md5	Especifica el algoritmo Hash con una política IKE.

6	**group** {**1** | **2** | **5**} **Ejemplo:** Router(config-isakmp-policy)# group 2	Especifica el identificador del grupo Diffie-Hellman con una política IKE
7	**encryption** {**des** | **3des** | **aes** | **aes 192** | **aes 256**} **Ejemplo:** Router(config-isakmp-policy)# encryption 3des	Especifica los algoritmos de encriptación con una política IKE.
8	**lifetime** seconds **Ejemplo:** Router(config-isakmp-policy)# lifetime 43200	Especifica el tiempo de vida de un S.A. Las herramientas del tiempo de vida del IKE son opcionales.
9	**Exit** **Ejemplo:** Router(config-isakmp-policy)# exit	Al ejecutar este comando se sale de la configuración de la política ISAKMP y se ingresa al modo de configuración global.
10	**crypto isakmp key** enc-type-digit keystring {**address** peer-address [mask] | **ipv6** {ipv6-address/ipv6-prefix} | **hostname** hostname} [**no-xauth**] **Ejemplo:** Router(config)# crypto isakmp key 0 my-preshare-key-0 address ipv6	Configura el intercambio de la autenticación de la llave.

	3ffe:1001::2/128	
11	**crypto keyring** keyring-name [**vrf** fvrf-name] **Ejemplo:** Router(config)# crypto keyring keyring1	Define una llave de encriptamiento que es utilizada durante la autenticación IKE.
12	**pre-shared-key** {**address** address [mask] | **hostname** hostname | **ipv6** {ipv6-address | ipv6-prefix}} **key** key **Ejemplo:** Router (config-keyring)# pre-shared-key ipv6 3FFE:2002::A8BB:CCFF:FE01:2C02/128	Define un intercambio de llave que es utilizado durante la autenticación del protocolo IKE.

9.4.1 Configurando IPsec de IPv6 en VTI (interfaz virtual del túnel)

Las siguientes tareas describen como configurar y habilitar IPsec de IPv6 en un túnel[73] virtual para IPv6.

Prerrequisitos

Utilizar el comando **ipv6 unicast-routing** para habilitar el ruteo unicast.

Pasos para implementar:

1. enable
2. configure terminal
3. ipv6 unicast-routing
4. interface tunnel *tunnel-number*

[73] *Ver capítulo 4 Redes Privadas Virtuales", sección 4.1 "Túneles"*

5. ipv6 address *ipv6-address/prefix*

6. ipv6 enable

7. tunnel source {*ip-address* | *ipv6-address* | *interface-type interface-number*}

8. tunnel destination {*host-name* | *ip-address* | *ipv6-address*}

9. tunnel mode {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **gre multipoint** | **gre ipv6** | **ipip** [**decapsulate-any**] | **ipsec ipv4** | **iptalk** | **ipv6** | **ipsec ipv6** | **mpls** | **nos** | **rbscp**}

10. tunnel protection ipsec profile *name* [**shared**]

A continuación se presentan los pasos en forma detallada

Paso	Comando	Propósito
1	**Enable** **Ejemplo**: Router> enable	Habilita el modo EXEC privilegiado.
2	configure Terminal **Ejemplo:** Router# configure terminal	Acceso al modo de configuración global,
3	**ipv6 unicast-routing** **Ejemplo:** Router(config)# ipv6 unicast-routing	Habilita el ruteo unicast de IPv6. Si solamente se necesita habilitar el ruteo unicast de IPv6, no importa la cantidad de interfaces del túnel se deseen configurar.
4	**interface tunnel** tunnel-number **Ejemplo:** Router(config)# interface tunnel 0	Especifica una interfaz del túnel y su número correspondiente, y entra al modo de la configuración de la interfaz.
5	**ipv6 address** ipv6-address/prefix **Ejemplo:** Router(config-if)# ipv6 address	Provee una dirección IPv6 a la interfaz del túnel, además el trafico IPV6 puede ser ruteado desde el túnel.

	3FFE:C000:0:7::/64 eui-64	
6	**ipv6 enable** **Ejemplo:** Router(config-if)# ipv6 enable	Habilita IPv6 en la interfaz del túnel.
7	**tunnel source** {ip-address | ipv6-address | interface-type interface-number} **Ejemplo:** Router(config-if)# tunnel source ethernet0	Establece la dirección origen para la interfaz del túnel.
8	**tunnel destination** {host-name | ip-address | ipv6-address} **Ejemplo:** Router(config-if)# tunnel destination 2001:0DB8:1111:2222::1	Especifica el destino para una interfaz del túnel.
9	**tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **gre multipoint** | **gre ipv6** | **ipip** [**decapsulate-any**] | **ipsec ipv4** | **iptalk** | **ipv6** | **ipsec ipv6** | **mpls** | **nos** | **rbscp**} **Ejemplo:** Router(config-if)# tunnel mode ipsec ipv6	Establece el modo de encapsulación para la interfaz del túnel. Para IPSec, solamente el comando **ipsec ipv6** es soportado
10	**tunnel protection ipsec profile** name [**shared**] **Ejemplo:** Router(config-if)# tunnel protection ipsec	Asocia a una interfaz del túnel con un archivo de IPsec. .

	profile profile1	

9.4.2 Verificando el modo de configuración del Protocolo Ipsec en modo túnel

Estas tareas son opciones y describen como presentar la información para verificar la configuración del protocolo Ipsec en modo túnel. Para ello, se utilizan los siguientes comandos que verifican la configuración y la operación[74]:

Pasos para implementar:

1. show adjacency [**summary** [*interface-type interface-number*]] | [*prefix*] [*interface interface-number*] [**connectionid** *id*] [**link** {**ipv4** | **ipv6** | **mpls**}] [**detail**]
2. show crypto engine {**accelerator** | **brief** | **configuration** | **connections** [**active** | **dh** | **dropped-packet** | **show**] | **qos**}
3. show crypto ipsec sa [**ipv6**] [*interface-type interface-number*] [**detailed**]
4. show crypto isakmp peer [**config** | **detail**]
5. show crypto isakmp policy
6. show crypto isakmp profile
7. show crypto map [**interface** *interface* | **tag** *map-name*]
8. show crypto session [**detail**] | [**local** *ip-address* [**port** *local-port*] | [**remote** *ip-address* [**port**
remote-port]] | [**detail**]] | [**fvfr** *vrf-name*] | [**ivrf** *vrf-name*]
9. show crypto socket
10. show ipv6 access-list [*access-list-name*]
11. show ipv6 cef [**vrf**] [*ipv6-prefix/prefix-length*] | [*interface-type interface-number*] [**longer-prefixes**
| **similar-prefixes** | **detail** | **internal** | **platform** | **epoch** | **source**]]
12. show interface *type number* **stats**

[74] *Ver capítulo 5 "Infraestructura de IPSec ", sección 5.9.1 "El protocolo ISAKMP"*

A continuación se presentan los pasos en forma detallada

Paso	Comando	Propósito
1	**show adjacency** [**summary** [interface-type interface-number]] | [prefix] [interface *interface-number*] [**connectionid** id] [**link** {**ipv4** | **ipv6** | **mpls**}] [**detail**] **Ejemplo:** Router# show adjacency detail	Despliega la información acerca de las tablas de adyacencia o de las tablas del hardware de la capa del nivel tres del modelo OSI.
2	**show crypto engine** {**accelerator** | **brief** | **configuration** | **connections** [**active** | **dh** | **dropped-packet** | **show**] | **qos**} **Ejemplo:** Router# show crypto engine connection active	Muestra un resumen de la información de la configuración del encriptamiento.
3	**show crypto ipsec sa** [**ipv6**] [interface-type interface-number] [**detailed**] **Ejemplo:** Router# show crypto ipsec sa ipv6	Despliega las herramientas utilizadas en el actual SA en IPv6.
4	**show crypto isakmp peer** [**config** | **detail**] **Ejemplo:** Router# show crypto isakmp peer detail	Muestra las descripciones de los lados que se ha implementado el protocolo ISAKMP.
5	**show crypto isakmp policy**	Despliega los parámetros para cada política IKE.

	Ejemplo: Router# show crypto isakmp policy	
6	**show crypto map** [**interface** interface | **tag** map-name] **Ejemplo:** Router# show crypto map	Muestra el mapa de configuración de crypto. El mapa crypto muestra los comandos de salidas que son generados dinámicamente.
7	**show crypto session** [**detail**] | [**local** ip-address [**port** local-port] | [**remote** ip-address [**port** remote-port]] | [**detail**]] | [**fvfr** vrf-name] | [**ivrf** vrf-name] **Ejemplo:** Router# show crypto session	Despliega la información del estado de la activaciones de las sesiones crypto.
8	**show crypto socket** **Ejemplo:** Router# show crypto socket	Lista de los sockets crypto.
9	**show ipv6 access-list** [access-list-name] **Ejemplo:** Router# show ipv6 access-list	Despliega el contenido de todas las actuales listas de acceso de IPv6.
10	**show ipv6 cef** [ipv6-prefix/prefix-length] | [interface-type interface-number] [**longer-prefixes** | **similar-prefixes** | **detail** | **internal** | **platform** | **epoch** | **source**]]	Despliega las entradas en la información de Reenvío de IPv6.

	Ejemplo: Router# show ipv6 cef	
11	**show interface** type number **stats** **Ejemplo:** Router# show interface fddi 3/0/0 stats	Despliega los números de los paquetes que fueron procesados, y distribuidos

Capítulo

10

Implementación de IPv6 con Linux y Windows

Temas:

- Prototipo de la red.
- Guía de Direccionamiento IPv6
- Guía de Instalación del protocolo IPv6 en Windows XP, Windows 2000 y Linux.
- Guía de Configuración Básica de IPv6 en Windows XP, Windows 2000 Y Linux
- Guía de configuración de servicios FTP, http en Linux
- Guía de implementación del Firewall en Linux.
- Guía de configuración de una VPN en linux.

10. Implementación del protocolo IPv6 en los sistemas operativos Linux y Windows.

En este capitulo se aborda la implementación del protocolo IPv6 en los sistemas operativos Linux Red Hat ES 4 y Windows XP. Para ello, se presenta diferentes instructivos que están detallados en forma teórica y ejemplos para su visualización. Las instrucciones se basan en un prototipo de red, que más adelante se detalla, en base a este modelo de red Local, se ha realizado la configuración adecuada para cada computadora.

Con la implementación se comprueba la teoría mencionada en los capítulos anteriores, se visualizan las características básicas como la configuración de una dirección IPv6, autoconfiguración, protocolo ICMP con el comando ping6, ip6tables, QoS, y seguridad. Estas características esenciales del protocolo se pueden comprobar no solo en routers sofisticados como CISCO[75], si no que también se verifican en servicios que proporciona el sistema Linux que simulan a un router.

10.1 Prototipo de la red

En el esquema siguiente, se visualiza al router, que permite la comunicación entre los usuarios remotos PC Cliente1 y PC Cliente2 con la LAN privada, donde se encuentran los usuarios PC Cliente3 y PC Cliente4. La red funciona bajo el esquema de ipv6 nativa, en la interfaz uno del router (tarjeta1) se encuentra la LAN interna, y en la interfaz 2 (tarjeta 2) se encuentra la conexión a Internet (que será simulada con otra red bajo el esquema de ipv6).

Los usuarios del extremo de Internet (Cliente1 y cliente2) se comunican hacia la Lan por medio de una VPN a través de IPsec. En el router que es el elemento de comunicación, será una computadora, configurada con el sistema operativo Linux, en la cual se configurará para que trabaje con firewall, QoS, VPN y primordialmente que tenga las funciones como router.

[75] *Ver capítulo 9 "Implementación de IPv6 con dispositivos CISCO"*

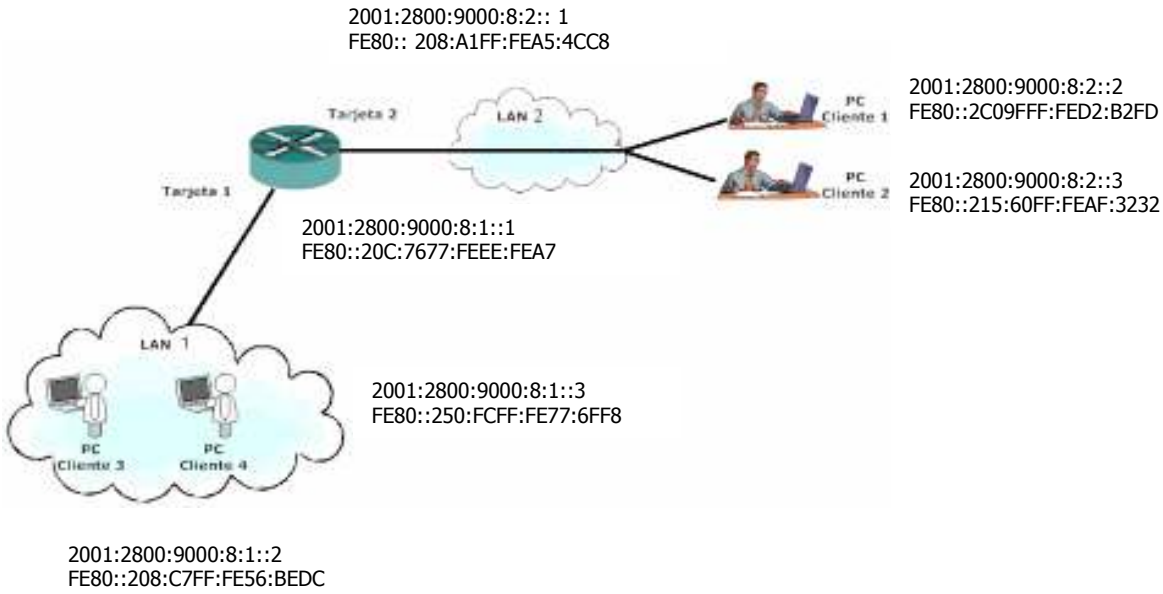


Figura 59. Esquema de la Red

10.1.1 Elementos a utilizar.

Se utilizan cinco computadoras y dos swithchs. Las computadoras estarán distribuidas de la siguiente forma:

Nombre	Dispositivo	Sistema Operativo	Ubicación
Router	Computadora 1	Linux, distribución Red Hat Enterprise ES 4	Dispositivo de comunicación entre la Lan y usuarios remotos
PC Cliente 1	Computadora 2	Microsoft Windows XP (SP 2)	Extremo de Internet
PC Cliente 2	Computadora 3	Microsoft Windows XP (SP 2)	Extremo de Internet
PC Cliente 3	Computadora 4	Microsoft Windows XP (SP 2)	LAN interna
PC Cliente 4	Computadora 5	Microsoft Windows Me (SP 2)	LAN interna

Tabla 8. Especificaciones de las computadoras.

Detalle de los Adaptadores de red

Nombre	N ° de tarjetas de red	Dirección MAC[76]
Router	**1 (eth0)**	00:0C:76:EE:FE:A7
Router	**2 (eth1)**	00:08:A1:A5:4C:C8
PC Cliente 1	**1**	00:C0:9f:D2:B2:Fd
PC Cliente 2	**1**	00:15:60:AF:32:32
PC Cliente 3	**1**	00-08-C7-56-BE-DC
PC Cliente 4	**1**	02:50:FC:77:6F:F8

Tabla 9 Direcciones MAC

El router posee dos interfaces (eth0 y eth1), las cuales dan soporte a redes diferentes. El cliente 1 y 2 pertenecen a la red 1, que da conectividad la interfaz eth0, y el cliente 3 y 4 pertenecen a la red2, que les brinda la conectividad la interfaz eth1. A continuación se detallan las direcciones ipv4 e ipv6 que se utilizaran.

1.) Direcciones a Utilizar con ipv4

[76] *La dirección MAC (Media Access Control Address) es un identificador hexadecimal de 48 bits, que corresponden de forma única con una tarjeta o interfaz de red.*

Red1 (eth0): 192.168.20.0/24

Red2 (eth1): 192.168.30.0/24

Nombre	Interfaz	Dirección
Router	Eth0	192.168.20.2
Router	Eth1	192.168.30.2
PC Cliente 1	4 (Link local área)	192.168.30.3
PC Cliente 2	4 (Link local área)	192.168.30.4
PC Cliente 3	4 (Link local área)	192.168.20.3
PC Cliente 4	4 (Link local área)	192.168.30.4

Tabla 10. Direcciones IPv4

2.) Direcciones a Utilizar con ipv6 (Direccionamiento de enlace local)

Red1: fe80:: /64

Red2: fe80:: /64

Nombre	Interfaz	Dirección
Router	Eth0	FE80::20C:7677:FEEE:FEA7
Router	Eth1	FE80::208:A1FF:FEA5:4CC8
PC Cliente 1	4 (Link local área)	FE80::2C09FFF:FED2:B2FD
PC Cliente 2	4 (Link local área)	FE80::215:60FF:FEAF:3232
PC Cliente 3	4 (Link local área)	FE80::208:C7FF:FE56:BEDC
PC Cliente 4	4 (Link local área)	FE80::250:FCFF:FE77:6FF8

Tabla 11 Direcciones Link Local.

2.) Direcciones a Utilizar con ipv6

Dirección de red: 2800:9000:8::/48

Red1: 2800:9000:8:0001::/64 equivalente a: 2800:9000:8:1::/64

Red2: 2800:9000:8:0002::/64 equivalente a: 2800:9000:8:2::/64

Nombre	Interfaz	Dirección
Router	Eth0	**2800:9000:8:1::1**
Router	Eth1	**2800:9000:8:2::1**
PC Cliente 1	4 (Link local área)	**2800:9000:8:2::2**
PC Cliente 2	4 (Link local área)	**2800:9000:8:2::3**
PC Cliente 3	4 (Link local área)	**2800:9000:8:1::2**
PC Cliente 4	4 (Link local área)	**2800:9000:8:1::3**

Tabla 12. Direcciones Globales

10.2 Guía de Direccionamiento IPv6

El direccionamiento de IPv6, fue explicado en el capítulo 2 “Introducción a IPv6”, sección 2.7 “Direccionamiento en IPv6”. Como un recordatorio, en la siguiente sección se explican los diferentes tipos de direcciones de IPv6, es necesario especificar, que una interfaz puede soportar varios tipos de direcciones IPv6, lo importante es saber el rango o nivel de alcance de la dirección. Con la siguiente retroalimentación, también se aclara la razón por la cual se utilizan las direcciones mencionadas en la sección 11.1.1 “Esquema de la red”.

A. Tipo de dirección “Link Local”

Son direcciones especiales, las cuales solamente son válidas en un enlace local de una interfaz. Si se utiliza este tipo de dirección como destino de un paquete, entonces, nunca va a poder pasar a través de un router. Este tipo de direccionamiento es utilizado para enlaces de comunicaciones tales como:

Existe otra computadora en este enlace?

Existe un dispositivo con una direccion especial?

Estas direcciones comienzan con (donde “X”es cualquier carácter hexadecimal, normalmente es “0”) [77]

fe8*x*: <- *Actualmente solo se utilize esta nomenclatura.*
fe9*x*:
fea*x*:
feb*x*:

Una dirección con este prefijo es buscada en cada interfaz IPv6 habilitada, después de una autoconfiguración de tipo staless.

[77] *Como ejemplo del tipo de direccionamiento Link Local, puede ver la tabla 11 de este capítulo.*

B. Tipo de dirección "Site Local"

Estas direcciones son similares a las que aparecen en la RFC 1978 ("Direcciones permitidas para redes privadas") es similar a las direcciones IPv4, y poseen la ventaja que si se utiliza este tipo de direccionamiento se obtienen 16 bits para un numero máximo de 65536 subnets. Otra ventaja es, que es posible asignar más de una dirección a una interfaz con IPv6, se puede asignar una para el sitio local y una a nivel global.

Estas direcciones comienzan con:

fec*x:* ***<- Es el comúnmente usado***
fed*x:*
fee*x:*
fef*x:*

(Donde "X", es cualquier character, nomalmente "0")

D. Direccionamiento Global

Las direcciones globales son definidas en la RFC 1884. Comienzan de la siguiente forma, donde la x puede ser cualquier tipo de carácter[78].

2*xxx*:
3*xxx*:

[78] *Como ejemplo puede ver la tabla 12, que muestra direcciones Globales a utilizar.*

10.2.1 Direcciones para los hosts

Para auto-configuración y movilidad, fue decidió utilizar los 64 bits inferiores para la parte del host de una dirección. Por lo tanto cada subset puede tener una dirección muy larga, La parte del host puede especificarse de la siguiente manera:

Automáticamente configurable (Staless)

Con auto-configuración, la parte del host de una dirección es configurada en base a la dirección MAC de la interfaz, con el método EUI-64[79], que hace que una dirección IPv6 sea única.

Ejemplo de direcciones MAC, para mayor ejemplo puede ver la tabla 9.

00:10:A4:E3:95:66

Primero, se convierte al formato EUI-64 insertando FF-FE entre el tercer y cuarto bytes, con el resultado de 00-10-A4-FF-FE-E3-95-66. Después, se complementa el séptimo bit del primer byte. El primer byte en formato binario es 00000000. Al complementar el séptimo bit, se convierte en 00000010 (0x02). El resultado final es 02-10-A4-FF-FE-E3-95-66 que, cuando se convierte a notación hexadecimal con dos puntos, da como resultado el identificador de interfaz 210:A4FF:FEE3:9566. En consecuencia, la dirección local del vínculo correspondiente al adaptador de red que tiene la dirección MAC de 00:10:A4:E3:95:66 es FE80::210:A4FF:FEE3:9566.

[79] *Ver capitulo 9 "Implementación de IPv6 con dispositivos CISCO "seccion 9.1 "Conectividad básica en IPv6"*

10.2.2 Prefijos Generales IPv6

Los 64 bits superiores de una dirección IPv6 están compuesto de un prefijo de ruteo global mas un subnet Id, Un prefijo general por ejemplo /48. Cuando un prefijo general es cambiado, todos los prefijos especificados basados ene le general también cambian, Esta función simplifica la renumeración de la red y permite automatizar la definición de los prefijos. Por ejemplo, un prefijo general puede tener 48 bits de longitud (48), y los demás prefijos generales puede tener 64 bits de longitud, en el siguiente ejemplo, a la izquierda hay 48 bits de todo los prefijos especificados, estos deben de ser iguales, los próximos 16 btis son todos diferentes.

Prefijo General: 2001:0DB8:2222::/48
Prefijo especifico: 2001:0DB8:2222:0000::/64
Prefijo especifico: 2001:0DB8:2222:0001::/64
Prefijo especifico: 2001:0DB8:2222:4321::/64
Prefijo especifico: 2001:0DB8:2222:7744::/64

Para la implementación del protocolo se utilizara el siguiente rango de direcciones:
2800:9000:8::/48

En la red1 (eth0) se utilizara:
2800:9000:8:1::/48

En la red2 (eth1) se utilizara:
2800:9000:8:2::/48

10.3 Guía de Instalación del protocolo IPv6 en Windows XP, Windows 2000 y Linux.

En general, las plataformas de Microsoft disponen de un buen soporte para IPv6. En la versión del sistema operativo "Windows XP", el protocolo viene preinstalado y su configuración es muy sencilla. En la página http://www.microsoft.com/IPv6 de Microsoft, aparece el soporte de la empresa al protocolo IPv6 en sus sistemas Operativos.

En Linux se implementa como un módulo del kernel. Así, las distribuciones con Kernel 2.2.x y 2.4.x ya vienen con este soporte y normalmente el módulo IPv6 ya está instalado. De todas formas, es importante asegurarse que el módulo se carga al arrancar. Es necesario hacer énfasis que este documento se basa en la distribución Red Hat ES 4. Una información detallada sobre el soporte IPv6 en las distribuciones más comunes se puede encontrar en: http://www.bieringer.de/linux/IPv6/status/IPv6+Linux-status-distributions.html

10.3.1 Instalación de IPv6 en plataforma Windows XP.

La instalación del protocolo IPv6 se puede realizar de dos maneras: la primera, desde la ventana de comandos, y la segunda desde el entorno de red.

10.3.1.1 Desde la ventana de comandos:

Pasos para la Instalación:

1. Instalar SP1 (Service Pack 1) de preferencia Superior (incluyendo Advanced Networking Pack para Windows XP).

2. Ingresar a la ventana de comandos (Clic en el menú inicio, Seleccionar ejecutar, luego escribir CMD, y presionar la tecla enter.

3. Digitar el comando "**IPv6 install**" o "**netsh interface IPv6 install** "desde el prompt de MS-DOS:

Posteriormente, aparecerá un mensaje indicando que ha sido correcta la instalación

Figura 60. Habilitando IPv6 en Windows XP.

Si se desea obtener una ayuda sobre el manejo y configuración de IPv6, entonces, siempre desde la ventana de comandos, puede ejecutar el comando IPv6 help, tal como se muestra en la figura 62, además puede ver el anexo 2 "Configuración de IPv6 en Windows XP"

Figura 61. El comando **IPv6 help**, se utiliza para desplegar la ayuda.

5. Para verificar la instalación fue correcta, puede revisar las propiedades de la conexión de la red, para acceder a este componente puede seguir los siguientes pasos:

- Inicio.
- Panel de Control.
- Conexión de Redes e Internet.
- Conexiones de Redes.
- Clic derecho al icono de conexión del área local

Luego aparecerá una ventana, como la de la figura 63:

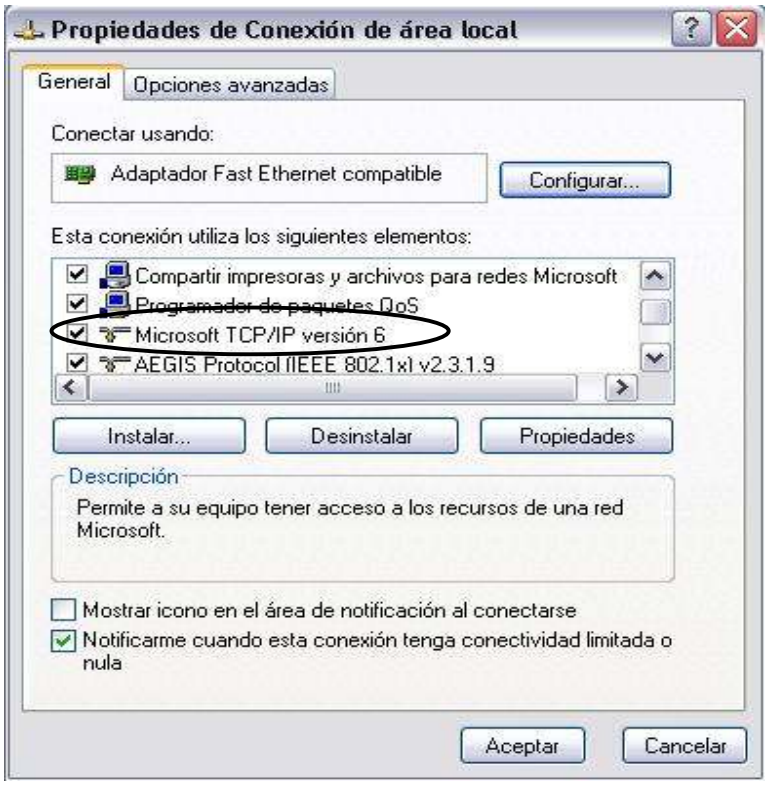


Figura 62 Verificación de la instalación del protocolo IPv6.

10.3.1.2 Desde el entorno de red:

Otra forma de agregar el protocolo IPv6 a Windows XP, es desde el entorno de comandos, para ingresar, de un clic en el menú inicio, luego dé un clic derecho sobre el icono de Mis sitios de red; del menú contextual que aparece seleccione propiedades e inmediatamente aparecerá una ventana como la de la figura 64.

Figura 63. Propiedades de Conexión

Luego presione el botón <**Instalar**>, e inmediatamente se cargará en pantalla una figura como la que se muestra a continuación:

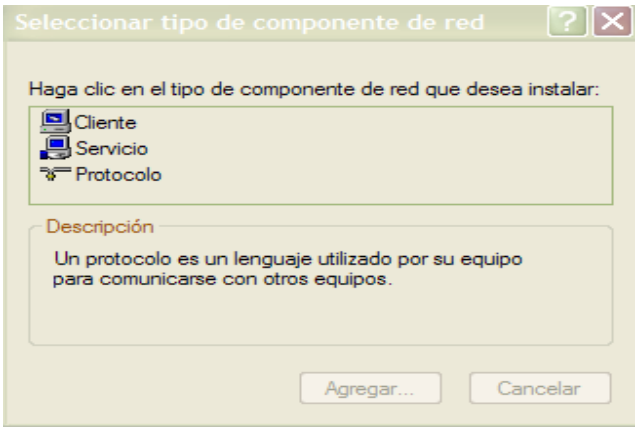


Figura 64. Instalación del protocolo IPv6

De la ventana anterior, se selecciona la opción Protocolo, y posteriormente se carga la siguiente ventana, de la cual se selecciona Microsoft TCP/IP versión 6. Finalmente de un clic en el botón <**Aceptar**>

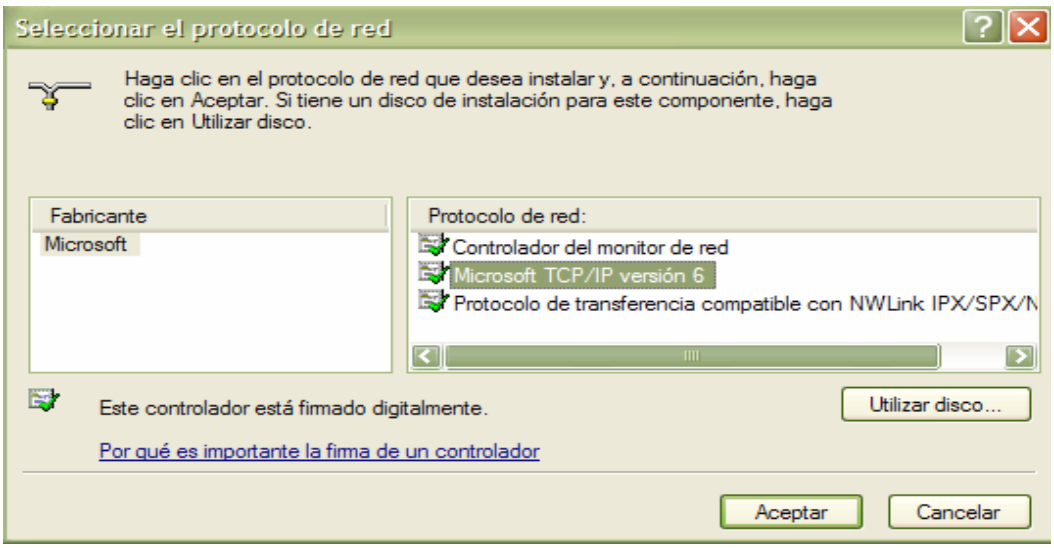


Figura 65.Se debe de especificar el protocolo IPv6

De igual manera, como la primera forma de instalar el protocolo, se puede verificar si fue instalado correctamente.

Figura 66 Verificación del protocolo IPv6.

10.3.2 Instalación de IPv6 en plataforma Windows 2000.

Windows 2000 también soporta el protocolo ipv6, en esta sección se explica la instalación y configuración en el sistema Windows 2000 SP 2.

Instalación de IPv6

- Es necesario descargar y abrir el fichero zip disponible en el siguiente enlace: https://noc.sixxs.net/archive/windows/
- Luego se descarga el archivo tpipv6-001205-SP2-IE6.zip
- Posteriormente se descomprime en una carpeta local, por ejemplo en C:\IPv6TP.
- Se procede a ejecutar, desde esa carpeta el fichero setup.exe. Es necesario reiniciar la PC.
- Desde el escritorio, se utiliza el botón derecho del ratón sobre el icono de Red, y se selecciona Propiedades, y de nuevo con el botón derecho sobre la tarjeta de red en la cual quieres instalar IPv6, selecciona Propiedades.
- Se debe de seleccionar Instalar, y después Protocolo y Agregar. Seleccione Microsoft IPv6 y finalmente Aceptar.

10.3.3 Instalación de IPv6 en plataforma Linux Red HAT ES 4.

El router tiene el sistema operativo Linux Red Hat ES 4. Por lo tanto, a continuación se detalla los pasos para verificar si el módulo de IPv6 se encuentra cargado correctamente. En este tipo de distribución, no se necesita agregar nada extra, solo verificar que las librerías estén correctamente cargadas, para otras distribuciones de Linux o kernel menor a 2.4 puede consultar el siguiente enlace http://www.bieringer.de/linux/IPv6/.

10.3.3.1 Comprobando el soporte del kernel para IPv6

1. Para verificar, si el kernel actual soporta IPv6, se verifica en /proc-file-system. La siguiente entrada:

```
[root@Root ~]# test -f /proc/net/if_inet6 && echo "Se esta ejecutando ipv6"
Se esta ejecutando ipv6
```

Si esto falla, significa que el módulo no esta cargado. Entonces se debe de cargar con la siguiente instrucción:

```
[root@Root ~]# modprobeipv6
```

Y si se desea verificar que la instalación fue satisfactoria, se digita lo siguiente:

```
[root@Root ~]# lsmod |grep –w 'ipv6' && echo "El modulo IPV6 fue cargado satisfactoriamente"
El modulo IPV6 fue cargado satisfactoriamente
```

10.3.3.2 Comprobando las herramientas de red para IPv6

El paquete de net-tools (herramientas de red), incluye algunas herramientas como ifconfig y route, las cuales ayudan a configurar IPv6 en una interfaz.

Se comprueba que la utilidad Ifconfig, que se utiliza para configurar la información necesaria en las tarjetas de red exista:

```
[root@Root ~]# /sbin/ifconfig -? 2>& 1|grep -qw 'inet6' && echo "La utilidad Ifconfig para Ipv6 esta lista"
La utilidad Ifconfig para Ipv6 esta lista
```

Se comprueba que la utilidad route, que se utiliza para configurar la información de las rutas exista:

```
[root@Root ~]# /sbin/route -? 2>& 1|grep -qw 'inet6' && echo  "La utilidad  route para Ipv6 esta lista"
La utilidad  route para Ipv6 esta lista
```

Linux posee una herramienta que configura las redes a través de las interfaces, que es inet6 la cual maneja el protocolo ip. Para comprobar su existencia, se ingresa en la consola lo siguiente:

```
[root@Root ~]# /sbin/ip 2>& 1 |grep -qw 'inet6' && echo "La utilidad ip para Ipv6 esta lista”
La utilidad ip para Ipv6 esta lista
```

Es importante mencionar que el router, posee dos tarjetas de red, y cada una de ella representa una subred, por lo tanto, se necesita activar el forwarding, para que el tráfico fluya de una tarjeta a otra.

Para verificar el estado del forwarding de ipv6 se utiliza lo siguiente:

```
[root@Root ~]# cat /proc/sys/net/ipv6/conf/all/forwarding
0
```

Con la siguiente instrucción se le asigna el valor de uno al forwarding.

```
[root@Root ~]# echo "1" >/proc/sys/net/ipv6/conf/all/forwarding
[root@Root ~]# cat /proc/sys/net/ipv6/conf/all/forwarding
1
```

En este caso, cuando se reinicia la computadora, pierde el valor de uno y queda de Nuevo a cero. Para evitar este inconveniente, se agrega “echo "1" >**/proc/sys/net/ipv6/conf/all/forwarding**” en el archivo rc.local. Así como se muestra en la figura 68.

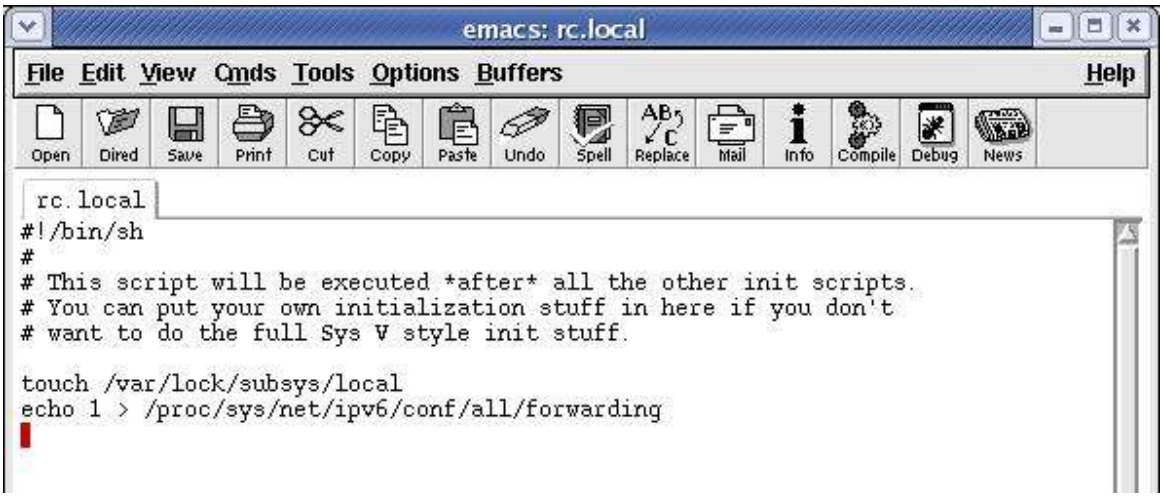


Figura 67. Activando el forwarding en el archivo rc.local

Cuando se reinicia la computadora, se puede verificar el valor del forwarding, de la siguiente manera:

```
[root@Root ~]# sysctl net.ipv6.conf.all.forwarding
net.ipv6.conf.all.forwarding = 1
```

10.3.3.3 Verificando el soporte de los scripts

Se puede verificar, si la distribución de Linux contiene el soporte adecuado de la configuración de la red para IPv6.

```
[root@Root ~]# test -f /etc/sysconfig/network-scripts/network-functions-ipv6 && echo "La principal librería IPv6 existe"
La principal librería IPv6 existe
```

Si se desea verificar la versión de la librería de la herramienta para el funcionamiento de red de ipv6, se introduce lo siguiente:

```
[root@Root ~]# source /etc/sysconfig/network-scripts/network-functions-ipv6 &&
getversion_ipv6_functions
20040321
```

En la distribución Red Hat ES 4, existe un directorio que se llama "/usr/share/doc/initscripts-7.93.11EL" que contiene el archivo "**sysconfig.txt**". El cual detalla las configuraciones necesarias para el funcionamiento de IPv6.

Figura 68. Directorio donde se encuentra la configuración de IPv6

Ejemplo del contenido del archivo, es el siguiente:

```
Files in /etc/sysconfig
=======================
/etc/sysconfig/authconfig
Used by authconfig to store information about the system's user information and
authentication setup; changes made to this file have no effect until the next time authconfig
is run.
/etc/sysconfig/network:
  NETWORKING_IPV6=yes|no
    Enable or disable global IPv6 initialization
    Default: no
  IPV6FORWARDING=yes|no
    Enable or disable global forwarding of incoming IPv6 packes on all interfaces.
    Note: Actual packet forwarding cannot be controlled per-device, use netfilter6 for such
issues
    Default: no
  IPV6INIT=yes|no
    Enable or disable IPv6 configuration for all interfaces
    Use with caution!
```

Default: value not set in this file
IPV6_AUTOCONF=yes|no
Sets the default for device-based autoconfiguration.
Default: yes if IPV6FORWARDING=no, no if IPV6FORWARDING=yes
IPV6_ROUTER=yes|no
Sets the default for device-based Host/Router behaviour.
Default: yes if IPV6FORWARDING=yes, no if IPV6FORWARDING=no

Se recomienda primero realizar el proceso de verificación y si se detecta algún problema se debe de revisar el archivo ***sysconfig.txt.*** *En el caso de Red Hat ES 4, no es necesario colocar nada extra en el archivo /etc/sysconfig/network, o la configuración de cada interfaz, por que ya posee por defecto las opciones de ipv6.*

10.4 Guía de verificación de la instalación en Windows XP, Windows 2000 y Linux.

Una vez instalado el protocolo IPv6, se debe de verificar si la configuración fue exitosa. En este caso, se comprueba tanto para Windows como para Linux, en ambos sistemas operativos, es diferente. Para la comprobación se utiliza la dirección loopback que es la **::1**, su equivalente en ipv6 es 127.0.0.1

10.4.1 En plataforma Windows 2000 y XP

Tanto para Windows 2000 y XP, se puede comprobar de la misma forma, se ejecuta desde la consola el siguiente comando:

```
C:\>ping6 ::1

Haciendo ping ::1
de ::1 con 32 bytes de datos:

Respuesta desde ::1: bytes=32 tiempo<1m
Respuesta desde ::1: bytes=32 tiempo<1m
Respuesta desde ::1: bytes=32 tiempo<1m
Respuesta desde ::1: bytes=32 tiempo<1m

Estadísticas de ping para ::1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

*Si el resultado no es satisfactorio, desinstale correctamente el protocolo desde la ventana de comandos, con la instrucción "i**pv6 uninstall**": y vuélvalo a instalar. Después de cada proceso es recomendable reiniciar el equipo.*

10.4.2 En plataforma Linux Red HAT ES 4.

Al igual que en las plataformas de Windows XP y Windows 2000, se comprueba la instalación del protocolo a través de la dirección loopback que es la **::1**

```
 [root@Root ~]# ping6 ::1
PING ::1(::1) 56 data bytes
64 bytes from ::1: icmp_seq=0 ttl=64 time=0.082 ms
64 bytes from ::1: icmp_seq=1 ttl=64 time=0.080 ms
64 bytes from ::1: icmp_seq=2 ttl=64 time=0.081 ms
```

10.5 Guía de configuración Básica de IPv6 en Windows XP, Windows 2000 Y Linux.

Una vez se instala y se comprueba el funcionamiento del protocolo, se procede a configurar los dispositivos, independiente del sistema operativo, para que el host pertenezca a una red, se le debe de configurar una dirección IPv6 y un gateway (o puerta de enlace), esto es en el caso de los pc's clientes. Para el caso del router (que es una computadora que trabaja con el sistema operativo Red Hat ES vs 4), no solo basta con realizar la configuración, también es necesario activar el forwarding para que se pueda tansmitir la información de una interfaz hacia otra.

Para windows XP, a parte de los comandos básicos para configurar IPv6, también se puede configurar a través de la herramienta netsh, mientras que en Windows 2000, el protocolo se maneja con la configuración básica y no se puede con netsh, lo que implica que existe una limitante en los comandos de configuración y verificación del protocolo.

Según el esquema a implementar las computadoras: pc1, pc2 y pc3 poseen el sistema Windows XP, mientras que la computadora pc4 tiene la plataforma de Windows 2000. La computadora central que hace la función de router, posee Linux Red Hat ES 4.

10.5.1 En plataforma Windows XP

En esta sección se describen las técnicas y las herramientas que se puede utilizar como ayuda para identificar un problema en capas sucesivas de la pila del Protocolo de control de transporte/Protocolo Internet (TCP/IP) que utiliza una capa Internet del Protocolo de Internet versión 6 (IPv6) en Microsoft Windows XP Para obtener la configuración IPv6 de un equipo, abra Símbolo del sistema. (Clic en el menú inicio, seleccione ejecutar, luego escriba **cmd**, y luego aceptar).

En Windows XP existe un software que se llama NETSH, el cual es un programa de la línea de comandos que permite, de forma local o remota, mostrar o modificar la configuración de red de un equipo que está en funcionamiento. Netsh también proporciona una característica de secuencias de comandos que permite ejecutar un conjunto de comandos en lotes en un equipo especificado.

Windows XP, 2003 y vista, utilizan Los comandos Netsh para la interfaz IPv6 proporcionan una herramienta de línea de comandos que puede utilizar para consultar y configurar interfaces IPv6, direcciones, cachés y rutas. Mientras que en Windows 2000 solo se permite la configuración básica.

Una recomendación para la configuración en computadoras Windows ya sea XP o Me, después de haber instalado el protocolo, es necesario volver a reiniciar la computadora, la razón se debe a que la computadora necesita establecer los identificadores de las interfaces correctamente.

En Windows XP, 2003 y Vista para tener control administrativo sobre la precedencia de las direcciones fuente/destino existe una tabla local de políticas de prefijos que se puede configurar como se muestra a continuación:

a) netsh interface ipv6 show prefixpolicy: muestra la tabla local de políticas de prefijos

C:\>netsh interface ipv6 show prefixpolicy
Consultando el estado activo...

Precedencia Etiq. Prefijo
---------- ----- ------------------------------
5 5 2001::/32
10 4 ::ffff:0:0/96
20 3 ::/96
30 2 2002::/16
40 1 ::/0
50 0 ::1/128

b) netsh interface ipv6 add prefixpolicy: añade nuevas entradas a la tabla local de políticas de prefijos

c) netsh interface ipv6 set prefixpolicy: configura entradas en la tabla local de políticas de prefijos

d) netsh interface ipv6 delete prefixpolicy: borra entradas en la tabla local de políticas de prefijos

Se puede utilizar las tareas siguientes para solucionar problemas relacionados con la conectividad IPv6:

- Comprobar la configuración
- Administrar la configuración
- Comprobar la no disponibilidad
- Ver y administrar la tabla de enrutamiento de IPv6
- Comprobar la confiabilidad del enrutador

- ***Comprobar la configuración***

Para comprobar si la configuración actual de IPv6 tiene la dirección correcta (cuando se configura manualmente) o una configuración de dirección apropiada (cuando se configura automáticamente) puede utilizar lo siguiente:

ipconfig /all: la presentación del comando **ipconfig /all** incluye direcciones IPv6, enrutadores predeterminados y servidores DNS para todas las interfaces. La herramienta Ipconfig sólo funciona en el equipo local.

netsh interface ipv6 show address: este comando sólo muestra las direcciones IPv6 asignadas a cada interfaz. En el siguiente ejemplo, se muestra el resultado de este comando:

C:\>netsh interface ipv6 show address
Consultando el estado activo...

- ***Administrar la configuración***

Ping6: se utiliza para verificar la conexión
ping6 [-t] [-a] [-n cuenta] [-l tamaño] [-w tiempo_espera] [-s dirección_origen] [-r] dest
Opciones:
-t: Ping de host especificado hasta interrumpirlo.
-a: Resolver direcciones para nombres de host.
-n: cuenta (Número de solicitudes de eco para mandar).
-l: tamaño (Enviar tamaño de búfer).
-w: tiempo_espera (Tiempo de espera en milisegundo para cada respuesta).
-s: dirección_origen (Dirección de origen).
-r: Usar encabezado de rutina para comprobar también la ruta contraria.

netsh interface ipv6 add address: se utiliza para configurar manualmente direcciones. Los parámetros de la sintaxis son los siguientes:

Etiqueta	Valor
interface	Nombre de interfaz o de índice.
address	Dirección IPv6 para agregarse.
type	Uno de los siguientes valores: A **unicast**: agrega una dirección de unidifusión (predeterminado). B. **anycast**: agrega una dirección de cualquier difusión.
validlifetime	Tiempo durante el que es válida la dirección. El valor predeterminado es infinito.
preferredlifetime	Tiempo durante el cual la dirección es preferida. El valor predeterminado es igual a validlifetime.
store	Uno de los siguientes valores: A. **active**, el cambio dura hasta el reinicio siguiente. B. **persistent**, el cambio persiste (predeterminado).

Ejemplo:

Netsh interface ipv6 add address 4 fec0::2:2c0:9fff:fed2:b2fd

netsh interface ipv6 set interface: se utiliza para realizar cambios en la configuración de las interfaces IPv6. Los parámetros de la sintaxis son los siguientes:

Etiqueta	Valor
interface	Nombre de interfaz o de índice.
forwarding	Si los paquetes llegados a esta interfaz pueden enviarse a otras interfaces. El predeterminado está deshabilitado.
advertise	Si los anuncios de enrutador son para mandarse en esta interfaz. El predeterminado está deshabilitado
mtu	El MTU de esta interfaz. El predeterminado es el MTU natural del vínculo.
siteid	Identificador de zona de ámbito de lugar.
metric	Interfaz métrica, agregada a rutas métricas para todas las rutas en la interfaz.
firewall	Si se desea operar en modo firewall.

siteprefixlength	Longitud predeterminada del prefijo global para todo el sitio.
store	Uno de los siguientes valores: **active,** el cambio durará hasta el reinicio siguiente. **persistent,** el cambio persiste (predeterminado).

Ejemplo:
Netsh interface ipv6 set address 4 fec0::2:2c0:9fff:fed2:b2fd store active

Netsh interface ipv6 delete address: se utiliza para eliminar una dirección. Los parametros que posee esta sintaxis son los siguientes:
Interface: Nombre de interfaz o de índice.
Address: Dirección IPv6 a borrar.
store : Uno de los siguientes valores:
active, la eliminación durará hasta el reinicio siguiente.
persistent, la eliminación persiste (predeterminado).

Ejemplo:
Netsh interface ipv6 delete address 4 fec0::2:2c0:9fff:fed2:b2fd

ipv6 if: se utliza para obtener el índice de la interfaz a través de la que se puede llegar a las direcciones del prefijo de ruta.

- ***Comprobar la no disponibilidad***

Para comprobar la no disponibilidad de un destino local o remoto, se verifican los siguientes comandos:

A. Verificando y vaciando la caché de vecinos

De manera similar a la caché del Protocolo de resolución de direcciones (ARP), la caché de vecinos almacena las direcciones de la capa de vínculo resueltas recientemente. Para ver el contenido actual de la caché de vecinos, utilice el comando **netsh interface ipv6 show neighbors**. Para vaciar la caché de vecinos, utilice el comando **netsh interface ipv6 delete neighbors**.

ipv6 nc: se utiliza para ver la entrada del caché de vecinos, utiliza el protocolo ND [80]

ipv6 rc: se utiliza para ver la entrada del caché de enrutamiento

ipv6 rt: para ver las entradas de la tabla de enrutamiento. Todos los equipos que ejecutan IPv6 determinan cómo deben reenviar los paquetes en función del contenido de la tabla de enrutamiento IPv6. Las entradas de la tabla de enrutamiento IPv6 constan de los elementos siguientes:

1. Un prefijo de dirección
2. La interfaz a través de la cual se envían los paquetes que coinciden con el prefijo de dirección.
3. Una dirección de reenvío o salto siguiente.
4. Un valor de preferencia que se utiliza para seleccionar entre varias rutas que tengan el mismo prefijo.
5. La duración de la ruta.
6. La especificación de si la ruta está publicada.
7. La especificación de caducidad de la ruta.
8. El tipo de ruta.

La tabla de enrutamiento IPv6 se genera automáticamente y está basada en la configuración IPv6 actual del equipo. Al reenviar los paquetes IPv6, el equipo busca en la tabla de enrutamiento la entrada más similar a la dirección IPv6 de destino. La ruta para el prefijo local del vínculo (FE80::/64) no se muestra.

La ruta predeterminada (una ruta con un prefijo de ::/0) se suele utilizar para reenviar un paquete IPv6 a un enrutador predeterminado del vínculo local. Como el enrutador que corresponde al enrutador predeterminado contiene información acerca de los prefijos de red de las demás subredes IPv6 del conjunto de redes IPv6 mayor, reenvía el paquete a otros enrutadores hasta que finalmente se entrega en el destino.

[80] *Ver capitulo 6 "Filtros de seguridad", sección 6.2 "ICMPv6"*

B. Verificando y limpiando la caché de destinos

La caché de destinos almacena las direcciones IPv6 de siguiente salto para los destinos. Para ver el contenido actual de la caché de destinos, utilice el comando **netsh interface ipv6 show destinationcache**. Para vaciar la caché de destinos, utilice el comando **netsh interface ipv6 delete destinationcache**

C Hacer ping al enrutador predeterminado

Utilice la herramienta Ping6 para hacer ping al enrutador predeterminado según su dirección IPv6. Puede obtener la dirección IPv6 de vínculo local del enrutador predeterminado de la información que muestran los comandos **ipconfig**, **netsh interface ipv6 show routes**, **route print** o **nbtstat -r**. Al hacer ping al enrutador predeterminado comprueba si puede alcanzar nodos locales y el enrutador predeterminado, que reenvía los paquetes IPv6 a nodos remotos.

Cuando se hace ping6 al enrutador predeterminado, debe especificar el identificador (Id.) de zona para la interfaz en la que desea que se envíen los mensajes Echo Request (Solicitud de eco) de ICMPv6. El identificador de la zona, es el índice de interfaz de la ruta predeterminada (::/0) con la métrica menor según muestran los comandos **netsh interface ipv6 show routes** o **route print**.

D. Se debe hacer ping a un destino remoto por su dirección IPv6

Si se puede hacer ping al enrutador predeterminado, se debe hacer ping a un destino remoto por la dirección IPv6.

E. Se hace un seguimiento de la ruta al destino remoto

Si no se puede hacer ping a un destino remoto por su dirección IPv6, quizás haya un problema de enrutamiento entre su nodo y el nodo de destino. Se utilice el comando **tracert6 –d** *DirecciónIPv6* para hacer un seguimiento de la ruta de enrutamiento hasta el destino remoto. La sintaxis completa del comando **tracert6** es la siguiente:

tracert6 [-d] [-h saltos_máximos] [-w tiempo_de_espera] [-s dirección_destino] nombre_destino

Opciones:

-d: No convierte direcciones en nombres de hosts.

-h: saltos_máximos (Máxima cantidad de saltos en la búsqueda del objetivo).

-w: tiempo_de_espera (Cantidad de milisegundos entre intentos).

-s dirección_destino

-r: Usar encabezado de rutina para comprobar también la ruta contraria.

Para una mayor comprensión, sobre los comandos básicos de la conectividad de IPv6 en el sistema operativo Windows, se presenta la configuración para las computadoras clienetes del prototipo de Red local.

Configurando las computadoras PC1, PC2 y PC3

Para configurar las computadoras que tienen el sistema Operativo Windows XP, se seguirá los siguientes pasos:

a) Para la implementación del esquema propuesto presentado en el inicio del capítulo, las computadoras tienen la conexión del área local en la interfaz 4, esto se logra percatar con la instrucción **ipv6 if 4**. Si se tiene duda que el indicador de la interfaz sea el correcto, solamente ingrese **ipv6 if**, y aparecerán de inmediato todas las conexiones que posea el ordenador. Cuando se digita **ipv6%numero_de_interfaz**, aparece lo siguiente:

D:\>ipv6 if 4
Interfaz 4: Ethernet: Conexión de área local
GUID {661A1860-F92F-42F7-9361-D2AAF6BB9B18}
usa descubrimiento de vecinos
usa descubrimiento de enrutador
dirección de capa de enlace: **00-c0-9f-d2-b2-fd**
preferred link-local **fe80::2c0:9fff:fed2:b2fd**, duración infinite
multidifusión interface-local ff01::1, 1 referencias , no reportable
multidifusión link-local ff02::1, 1 referencias , no reportable
multidifusión link-local ff02::1:ffd2:b2fd, 1 referencias , último informado
multidifusión link-local ff02::1:ff00:2, 1 referencias , último informador
enlace MTU 1500 (enlace MTU 1500)
límite de saltos actual128
tiempo alcanzable 27500ms (base 30000ms)
intervalo de retransmisión 1000ms
transmisiones DAD 1
longitud de prefijo de sitio predeterminada 48

En el ejemplo anterior se observa lo siguiente:

Interfaz 4: Ethernet: Conexión de área local

Indica el número de la interfaz, y a que tipo de conexión se refiere, en este caso, la interfaz 4 es la conexión de área local.

Usa descubrimiento de vecino

El descubrimiento de vecinos consiste en una serie de mensajes ICMPv6 que administran la comunicación de un host a otro en un vínculo, y determina las direcciones de otros hosts en el mismo enlace[81].

Usa descubrimiento de enrutador

Consiste en el uso de mensajes ICMPv6 para descubrir la puerta de enlace predeterminada de un segmento de red. El descubrimiento de enrutadores consta de dos mensajes ICMPv6: la solicitud de enrutador y el anuncio de enrutador.

Dirección de capa de enlace: 00-c0-9f-d2-b2-fd

Muestra la dirección MAC que es utilizada cuando se autoconfigura las direcciones ipv6.

Link-local fe80::2c0:9fff:fed2:b2fd

Esta es la dirección de enlace local, puede ver la tabla 11 de este capitulo.

Multidifusión interface-local ff01::1,

En una dirección de multidifusión se identifican varias interfaces. Con la topología de enrutamiento de multidifusión adecuada, los paquetes dirigidos a una dirección de multidifusión se entregan en todas las interfaces identificadas en ella. Se identifican por que comienza con FF. En este caso FF01::1 es la dirección para todos los nodos de ámbito local del nodo.

Multidifusión link-local ff02::1

Dirección para todos los nodos de ámbito local del vínculo

Multidifusión link-local ff02::1:ffd2:b2fd

[81] *Ver capítulo 6 "Filtros de seguridad", sección 6.2 "ICMPv6"*

Represente a la dirección de multidifusión del nodo, ff02::1 y ffd2:b2fd lo toma de la dirección MAC.

Multidifusión link-local ff02::1:ff00:2

En lugar de utilizar la dirección para todos los nodos de ámbito local del vínculo como destino del mensaje de solicitud de vecino, que afectaría a todos los nodos IPv6 del vínculo local, se utiliza la dirección de multidifusión de nodo solicitado ff02::1:ff00:2

b) Como segundo paso, se entra a la herramienta **Netsh,** luego se establece la **interface ipv6** y se utiliza el comando **add address identificador_interface dirección_ipv6**.

c) El tercer paso, es asignarle la dirección de la puerta de enlace determinada (gateway) para ello, siempre desde la herramienta **netsh**, se entra a **la interface ipv6**, y se coloca lo siguiente: **add route ::/0 identificador_interfaz dirección_gateway.**

d) Luego con **ipconfig**, se verifican las direcciones de enlaces, la global agregada manualmente, y la direccion del gateway.

e) Finalmente se hace **ping6** a la dirección de enlace local y a la global.

1. Configuración para PC1

Direcciones:

IPv6 (link local): **FE80::2C0:9FFF:FED2:B2FD**

IPv6 (global): **2800:9000:8:2::2**

Se verifica que el número de la interfaz 4 pertenezca a la conexión de área local.

```
D:\>ipv6 if 4
Interfaz 4: Ethernet: Conexión de área local
GUID {661A1860-F92F-42F7-9361-D2AAF6BB9B18}
usa descubrimiento de vecinos
usa descubrimiento de enrutador
dirección de capa de enlace: 00-c0-9f-d2-b2-fd
preferred link-local fe80::2c0:9fff:fed2:b2fd, duración infinite
multidifusión interface-local ff01::1, 1 referencias , no reportable
multidifusión link-local ff02::1, 1 referencias , no reportable
multidifusión link-local ff02::1:ffd2:b2fd, 1 referencias , último informado
multidifusión link-local ff02::1:ff00:2, 1 referencias , último informador
enlace MTU 1500 (enlace MTU 1500)
límite de saltos actual128
tiempo alcanzable 27500ms (base 30000ms)
intervalo de retransmisión 1000ms
transmisiones DAD 1
longitud de prefijo de sitio predeterminada 48
```

A través de la herramienta netsh, se le asigna la dirección global a la interfaz 4.

```
D:\>netsh
netsh>interface ipv6 add address 4 2800:9000:8:2::2
Aceptar
```

Se procede a asignarle la dirección del gateway (que es la dirección de la eth1 del router).

```
netsh>interface ipv6 add route ::/0 4 2800:9000:8:2::1
Aceptar
```

Para identificar, si se realizaron los cambios, se verifica a través de ipconfig.

```
D:\>ipconfig

Configuración IP de Windows

Adaptador Ethernet Conexión de área local        :

      Sufijo de conexión específica DNS :
      Dirección IP. . . . . . . . . . . : 192.168.30.3
      Máscara de subred . . . . . . . . : 255.255.255.0
      Dirección IP. . . . . . . . . . . : 2800:9000:8:2::2
      Dirección IP. . . . . . . . . . . : fe80::2c0:9fff:fed2:b2fd%4
      Puerta de enlace predeterminada   : 192.168.30.2
                                          2800:9000:8:2::1
```

Se debe de hacer ping6 a la dirección de enlace local.

```
D:\>ping6 fe80::2c0:9fff:fed2:b2fd

Haciendo ping fe80::2c0:9fff:fed2:b2fd
de fe80::2c0:9fff:fed2:b2fd%4 con 32 bytes de datos:

Respuesta desde fe80::2c0:9fff:fed2:b2fd%4: bytes=32 tiempo<1m
Respuesta desde fe80::2c0:9fff:fed2:b2fd%4: bytes=32 tiempo<1m
Respuesta desde fe80::2c0:9fff:fed2:b2fd%4: bytes=32 tiempo<1m
Respuesta desde fe80::2c0:9fff:fed2:b2fd%4: bytes=32 tiempo<1m

Estadísticas de ping para fe80::2c0:9fff:fed2:b2fd:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Para comprobar que la dirección global configurada manualmente es correcta se hace ping6 a la dirección.

```
D:\>ping6 2800:9000:8:2::2

Haciendo ping 2800:9000:8:2::2
de 2800:9000:8:2::2 con 32 bytes de datos:

Respuesta desde 2800:9000:8:2::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::2: bytes=32 tiempo<1m

Estadísticas de ping para 2800:9000:8:2::2:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

2. Configuración para PC2

Direcciones:

IPv6 (link local): **FE80::215:60FF:FEAF:3232**

IPv6 (global): **2800:9000:8:2::3**

Se verifica que el número de la interfaz 4 pertenezca a la conexión de área local.

```
C:\>ipv6 if 4
Interfaz 4: Ethernet: Conexión de área local
GUID {404178FF-348D-4B57-9134-8BAFF5D2EE2D}
usa descubrimiento de vecinos
usa descubrimiento de enrutador
dirección de capa de enlace: 00-15-60-af-32-32
preferred link-local fe80::215:60ff:feaf:3232, duración infinite
multidifusión interface-local ff01::1, 1 referencias , no reportable
multidifusión link-local ff02::1, 1 referencias , no reportable
multidifusión link-local ff02::1:ffaf:3232, 1 referencias , último informado
multidifusión link-local ff02::1:ff00:3, 1 referencias , último informador
enlace MTU 1500 (enlace MTU 1500)
límite de saltos actual128
tiempo alcanzable 25500ms (base 30000ms)
intervalo de retransmisión 1000ms
transmisiones DAD 1
longitud de prefijo de sitio predeterminada 48
```

A través de la herramienta netsh, se le asigna la dirección global a la interfaz 4.

```
C:\>netsh
netsh>interface ipv6 add address 4 2800:9000:8:2::3
Aceptar
```

Se procede a asignarle la dirección del gateway (que es la dirección de la eth1 del router).

```
netsh>interface ipv6 add route ::/0 4 2800:9000:8:2::1
Aceptar
```

Para identificar, si se realizaron los cambios, se verifica a través de ipconfig.

```
C:\>ipconfig

Configuración IP de Windows
Adaptador Ethernet Conexión de área local        :

      Sufijo de conexión específica DNS :
      Dirección IP. . . . . . . . . . . : 192.168.30.4
      Máscara de subred . . . . . . . . : 255.255.255.0
      Dirección IP. . . . . . . . . . . : 2800:9000:8:2::3
      Dirección IP. . . . . . . . . . . : fe80::215:60ff:feaf:3232%4
      Puerta de enlace predeterminada   : 192.168.30.2
                                          2800:9000:8:2::1
```

Se debe de hacer ping6 a la dirección de enlace local.

```
C:\>ping6 fe80::215:60ff:feaf:3232

Haciendo ping fe80::215:60ff:feaf:3232
de fe80::215:60ff:feaf:3232%4 con 32 bytes de datos:

Respuesta desde fe80::215:60ff:feaf:3232%4: bytes=32 tiempo<1m
Respuesta desde fe80::215:60ff:feaf:3232%4: bytes=32 tiempo<1m
Respuesta desde fe80::215:60ff:feaf:3232%4: bytes=32 tiempo<1m
Respuesta desde fe80::215:60ff:feaf:3232%4: bytes=32 tiempo<1m

Estadísticas de ping para fe80::215:60ff:feaf:3232:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Para comprobar que la dirección global configurada manualmente es correcta se hace ping6 a la dirección.

```
C:\>ping6 2800:9000:8:2::3

Haciendo ping 2800:9000:8:2::3
de 2800:9000:8:2::3 con 32 bytes de datos:

Respuesta desde 2800:9000:8:2::3: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::3: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::3: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::3: bytes=32 tiempo<1m

Estadísticas de ping para 2800:9000:8:2::3:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

3. Configuración para PC3

Direcciones:

IPv6 (link local): **FE80::208:C7FF:FE56:BEDC**

IPv6 (global): **2800:9000:8:1::2**

Se verifica que el número de la interfaz 4 pertenezca a la conexión de área local.

```
C:\>ipv6 if 4
Interfaz 4: Ethernet: Conexión de área local 2
GUID {D5A58B64-B62F-4423-A50E-55C766B28056}
usa descubrimiento de vecinos
usa descubrimiento de enrutador
dirección de capa de enlace: 00-08-c7-56-be-dc
preferred link-local fe80::208:c7ff:fe56:bedc, duración infinite
multidifusión interface-local ff01::1, 1 referencias , no reportable
multidifusión link-local ff02::1, 1 referencias , no reportable
multidifusión link-local ff02::1:ff56:bedc, 1 referencias , último informado
multidifusión link-local ff02::1:ff00:2, 1 referencias , último informador
enlace MTU 1500 (enlace MTU 1500)
límite de saltos actual128
tiempo alcanzable 36500ms (base 30000ms)
intervalo de retransmisión 1000ms
transmisiones DAD 1
longitud de prefijo de sitio predeterminada 48
```

A través de la herramienta netsh, se le asigna la dirección global a la interfaz 4.

```
C:\>netsh
netsh>interface ipv6 add address 4 2800:9000:8:1::2
Aceptar
```

Se procede a asignarle la dirección del gateway (que es la dirección de la eth1 del router).

```
netsh>interface ipv6 add route ::/0 4 2800:9000:8:1::1
Aceptar
```

Para identificar, si se realizaron los cambios, se verifica a través de ipconfig.

```
C:\>ipconfig

Configuración IP de Windows

Adaptador Ethernet Conexión de área local 2        :

      Sufijo de conexión específica DNS :
      Dirección IP. . . . . . . . . . . : 192.168.20.3
      Máscara de subred . . . . . . . . : 255.255.255.0
      Dirección IP. . . . . . . . . . . : 2800:9000:8:1::2
      Dirección IP. . . . . . . . . . . : fe80::208:c7ff:fe56:bedc%4
      Puerta de enlace predeterminada   : 192.168.20.2
                                          2800:9000:8:1::1
```

Se debe de hacer ping6 a la dirección de enlace local.

```
C:\>ping6 fe80::208:c7ff:fe56:bedc

Haciendo ping fe80::208:c7ff:fe56:bedc
de fe80::208:c7ff:fe56:bedc%4 con 32 bytes de datos:

Respuesta desde fe80::208:c7ff:fe56:bedc%4: bytes=32 tiempo<1m
Respuesta desde fe80::208:c7ff:fe56:bedc%4: bytes=32 tiempo<1m
Respuesta desde fe80::208:c7ff:fe56:bedc%4: bytes=32 tiempo<1m
Respuesta desde fe80::208:c7ff:fe56:bedc%4: bytes=32 tiempo<1m

Estadísticas de ping para fe80::208:c7ff:fe56:bedc:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Para comprobar que la dirección global configurada manualmente es correcta se hace ping6 a la dirección.

```
C:\>ping6 2800:9000:8:1::2

Haciendo ping 2800:9000:8:1::2
de 2800:9000:8:1::2 con 32 bytes de datos:

Respuesta desde 2800:9000:8:1::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::2: bytes=32 tiempo<1m

Estadísticas de ping para 2800:9000:8:1::2:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

10.5.2 En plataforma Windows 2000

Los pasos necesarios para asignarle una dirección ipv6 al nodo de Windows 2000, son similares que en Windows XP; con la diferencia que Windows 2000 tiene la limitante que no maneja el protocolo ipv6 a través de la herramienta Netsh, solo da acceso a una configuración básica. Las instrucciones a seguir son las siguientes:

a) Asegurarse que el indicador de la interfaz del área Local, sea el correcto, esto se logra digitando en la consola ipv6 if 4, si se tiene duda del numero de la interfaz, puede digitar **ipv6 if**, que le mostrará de inmediato todas las interfaces que existen en el ordenador.

b) Una vez identificada el identificador de la interfaz apropiada se procede a asignarle una dirección ipv6, se digita: ipv6 adu identificador_interfaz/direccion_ipv6.

c) Luego se procede a asignarle la dirección de la puerta de enlace (gateway)

Configuración para PC4

Direcciones:

IPv6 (link local): **FE80::250:fcff:fe77:6ff8**

IPv6 (global): **2800:9000:8:1::3**

Se introduce **ipv6 if 4** para verificar que el identificador 4 pertenece a la conexión del area local.

```
D:\>ipv6 if 4
Interface 4 (site 1): Local Area Connection
uses Neighbor Discovery
link-level address: 00-50-fc-77-6f-f8
preferred address fe80::250:fcff:fe77:6ff8, infinite/infinite
multicast address ff02::1, 1 refs, not reportable
multicast address ff02::1:ff77:6ff8, 1 refs, last reporter
link MTU 1500 (true link MTU 1500)
current hop limit 128
reachable time 41000ms (base 30000ms)
retransmission interval 1000ms
DAD transmits 1
```

A diferencia de Windows XP, en Windows 2000, se introduce la dirección ipv6 con el comando **adu**, se debe de especificar también el identificador de la interfaz.

```
D:\>ipv6 adu 4/2800:9000:8:1::3
```

En Windows 2000 ipv6 no se puede manejar a través de la herramienta netsh, por lo tanto la dirección del gateway, se ingresa con el comando **rtu**.

```
D:\>ipv6 rtu ::/0 4/2800:9000:8:1::1
```

Si se introduce ipconfig, no se puede visualizar las direcciones configuradas, entonces, se recomienda digitar de nuevo **ipv6 if 4.**

```
D:\>ipconfig

Windows 2000 IP Configuration

Ethernet adapter Local Area Connection:
        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.20.4
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.20.2
```

Se debe de hacer ping6 a la dirección de enlace local.

```
D:\>ping6 fe80::250:fcff:fe77:6ff8

Pinging fe80::250:fcff:fe77:6ff8 with 32 bytes of data:

Reply from fe80::250:fcff:fe77:6ff8%4: bytes=32 time<1ms
Reply from fe80::250:fcff:fe77:6ff8%4: bytes=32 time<1ms
Reply from fe80::250:fcff:fe77:6ff8%4: bytes=32 time<1ms
Reply from fe80::250:fcff:fe77:6ff8%4: bytes=32 time<1ms
```

Para comprobar que la dirección global configurada manualmente es correcta se hace ping6 a la dirección.

```
D:\>ping6 2800:9000:8:1::3

Pinging 2800:9000:8:1::3 with 32 bytes of data:

Reply from 2800:9000:8:1::3: bytes=32 time<1ms
Reply from 2800:9000:8:1::3: bytes=32 time<1ms
Reply from 2800:9000:8:1::3: bytes=32 time<1ms
Reply from 2800:9000:8:1::3: bytes=32 time<1ms
```

10.5.2 En plataforma Linux Red HAT ES 4.

La computadora que posee el sistema operativo Linux Red Hat ES 4, es la que se utiliza como un router. Para que tenga esa funcionalidad especial, se utiliza el sistema Quagga, el cual brinda el servicio de Zebra. Es importante mencionar, que en algunas versiones anteriores de Linux, era necesario descargarlo, y configurarlo, En el caso de Red Hat ES 4, el servicio ya se encuentra integrado al sistema operativo, más adelante se brindará los pasos a seguir para activarlo y configurarlo.

Linux, también tiene la opción de realizar la configuración básica desde Terminal, y algunos comandos son parecidos a los de Windows. La razón por la cual se utiliza Zebra es por que la computadora brinda las funciones de un router. En esta sección, se explican de las dos formas de configuración: Configuración como un router, y configuración con comandos desde el Terminal.

10.5.2.1 Configuración de Linux con Zebra.

Un sistema con Quagga instalado actúa como router dedicado. Con Quagga, una máquina intercambia información de routing con otros routers utilizando protocolos de routing. Quagga utiliza esa información para actualizar el núcleo de las tablas de routing de forma que la información correcta esté en el lugar correcto. Quagga permite la configuración dinámica y es posible ver la información de la tabla de routing desde el interfaz de terminal de Quagga.

Tradicionalmente, la configuración de un router basado en Linux se realizaba mediante los comandos ***ifconfig*** y los comandos del tipo ***route.*** El estado de las tablas se podía mostrar mediante la utilidad ***netstat***. Estos comandos solamente se podían utilizar trabajando como root. Quagga, sin embargo tiene otro método de administración. En Quagga existen dos modos de usuario. Uno es el modo normal y el otro es el modo de enable (habilitado). El usuario de modo normal únicamente puede ver el estado del sistema, sin embargo el usuario de modo enable puede cambiar la configuración del sistema.

Pasos para configurar zebra:

1. EL primer paso que se debe de efectuar, es colocar el nombre de la computadora (host), esto se hace a través del archivo **/etc/hosts**. En esta implementación el nombre del host es Root.

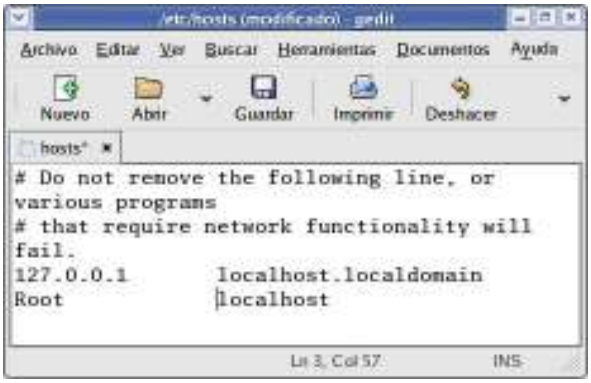


Figura 69. Se le asigna un hombre al host en el archivo /etc/hosts

2. Luego se verifica y modifica los siguientes archivos de configuración, que se encuentran en **/etc/quagga**:

- **Zebra.conf (archivo básico para quagga)**
- **Ripd.conf (archivo para manejar el protocolo RIP)**
- **Ripng.conf (archivo para manejar el protocolo Rip nueva genración, en el caso de IPv6)**

3. Si no se encuentran los archivos, entonces se deben de crear y guardarlos en el directorio /**etc/quagga**. Los archivos deben de contener como mínimo los comandos básicos de los archivos zebra.conf.sample, ripd.conf.sample, ripng.conf.sample, que se encuentran en el directorio /etc/quagga. Cuando se abre el archivo zebra.conf, se visualiza la siguiente información

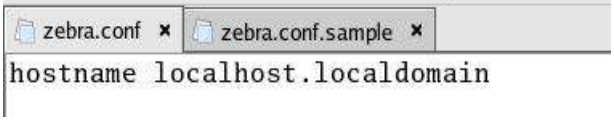


Figura 70. Contenido del archivo zebra.conf

El contenido del archivo de la figura 71, no es de mucha ayuda, se debe de borrar esa línea y tomar de ejemplo la configuración básica que aparece en el archivo zebra.conf.sample.

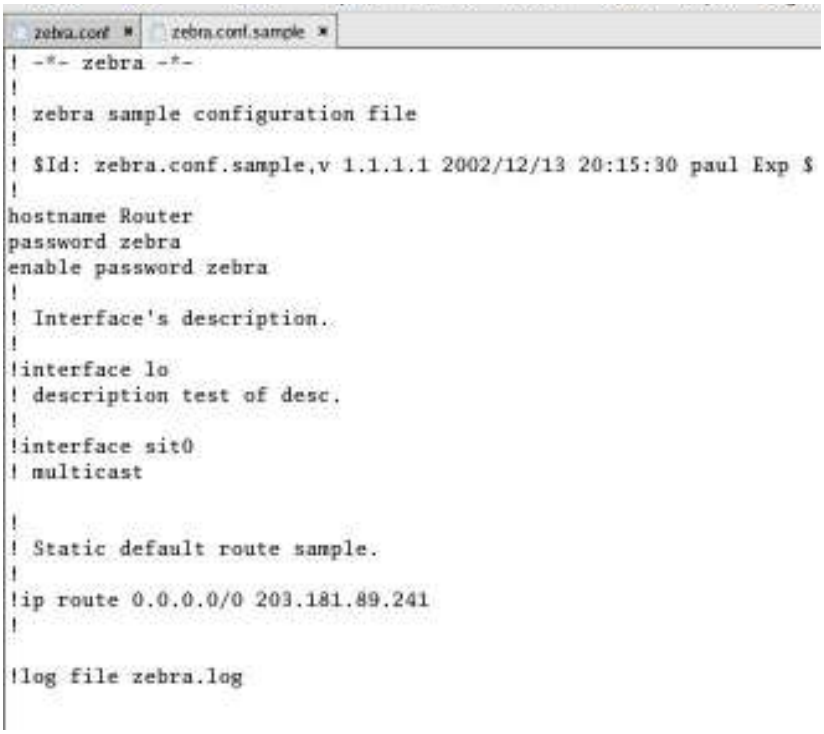


Figura 71. Archivo zebra.conf.sample

Se copia la configuración básica del archivo zebra.conf.sample al archivo zebra.conf. Así como aparece en la figura 73. La configuración básica consiste en:

Hostname Root

Password Zebra

Enabled password zebra

Log stdout

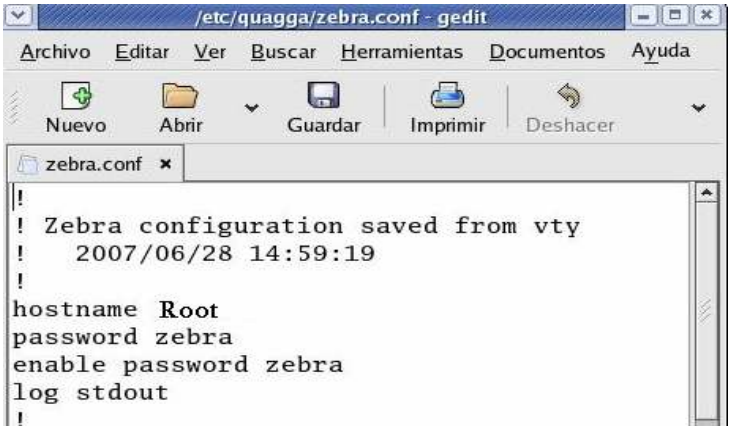


Figura 72. Configuración básica para el archivo zebra.conf

4. Posteriormente se debe de activar el servicio, para ello se ingresa al menú de aplicaciones y luego seleccione configuración del sistema, y escoja configuración de servidores y dé un clic en servicios. Así como s muestra en la figura 74.

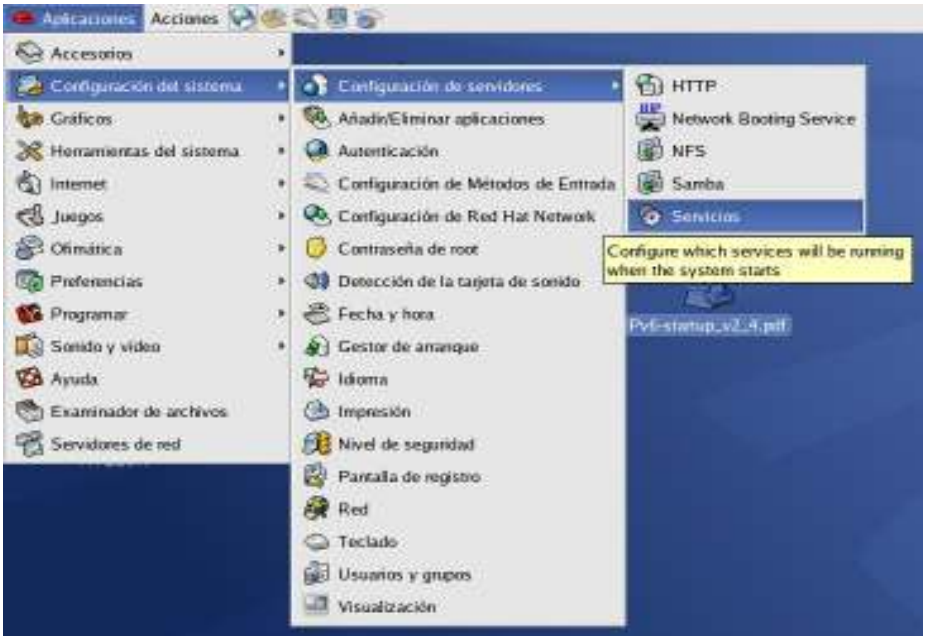


Figura 73. Ubicación para activar servicios.

5. Luego aparecerá una ventana, como la que se muestra en la figura 75, se selecciona el servicio: zebra.

Figura 74. Activación de servicios.

6. Se presiona el botón <**iniciar**>, para ejecutar el servicio e inmediatamente aparecerá un mensaje como el de la figura 76. Para que el servicio se inicie siempre se dá un clic en el botón <**guardar**>.

Figura 75. Mensaje de activación del servicio

El proceso de ejecución del servicio, también se puede realizar a través de la consola, se digita lo siguiente:

Zebra –d

Con la diferencia que cada vez que se reinicia la computadora es necesario digitar esta instrucción.

7. Para finalizar, se comprueba la configuración a través de la Terminal, para ello, se digita el comando **vtysh,** y aparecerá un mensaje de bienvenida y la versión del Quagga.

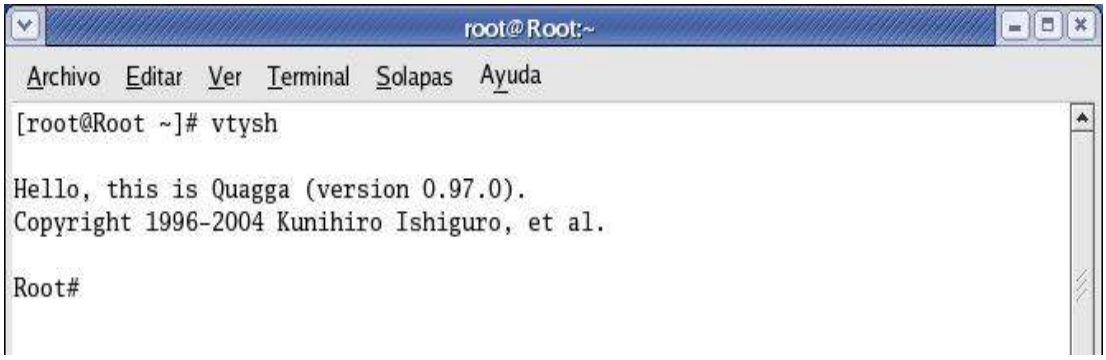


Figura 76. Verificación de la configuración de Zebra.

10.5.2.2 Pasos para configurar el router

1. Para realizar esta configuración, es necesario ingresar a la consola, y digitar el comando **vtysh**.
[root@Root ~]# vtysh

2. A continuación se presenta **la configuración para las interfaces eth0 y eth1**, que corresponde a la red1. Los comandos que se utilizan en el router zebra son en la mayoría similares a los aceptados por los dispositivos de CISCO. En la configuración siguiente se utilizan los comandos:

Paso	Comando	Propósito
1	Configure terminal	Accede al modo de configuración global.
2	Interface Eth0	Se especifica que la interfaz a configurar es la interfaz Eth0
3.	Ip address dirección/prefijo	Con este comando, se asigana la dirección Ip a la interfaz eth0.
4	Ipv6 address dirección/prefijo	Con este comando, se asigana la dirección IPv6 a la interfaz eth0.
5.	Description	El comando se utiliza, para colocar una

		breve descripción a la interfaz, que servirá como referencia.
6	No shutdown	Este comando es neceario para activar la interfaz.
7.	Exit	Ejecuta la salida del modo de configuración en una interfaz y regresa al modo de configuración global.

De acuerdo a los comandos anteriores, las configuraciones para la interfaz Eth0 y Eth1, son las siguientes:

Interfaz ETH0

root# conf t

root(config)# int eth0

Root(config-if)# ip address 192.168.20.2/24

root(config-if)# ipv6 address 2800:9000:8:1::1/64

root(config-if)# description NETWORK1

Root(config-if)# no shutdown

Root(config-if)# exit

Interfaz ETH1

Root# conf t

root(config)# int eth1

Root(config-if)# ip address 192.168.30.2/24

root(config-if)# ipv6 address 2800:9000:8:2::1/64

Root(config-if)# description NETWORK2

Root(config-if)# no shutdown

Root(config-if)# exit

3. Una vez se realice la configuración básica para cada interfaz, se debe de verificar la configuración. Existen muchos comandos para visualizar la configuración realizada, se pueden emplear los siguientes comandos: **show running-config, show ipv6 route.**

a. Show running-config: este comando se utiliza para verificar la configuración ingresada, presenta un resumen de la configuración, así se puede comprobar que las instrucciones que se hayan ingresado sean las correctas, con esta instrucción, se verifican claramente las direcciones IPv6 ingresadas para cada interfaz. Como se ha explicado anteriormente, las interfaces pueden soportor multiples direcciones IPv6. Por defecto en cada interfaz aparece una instrucción "ipv6 nd suppress-ra", que indica está habilitada la función de detectar a los vecinos en una red.

show running-config

Root# show running-config

Building configuration...

Current configuration:

!

hostname Router

password zebra

enable password zebra

log stdout

!

interface eth0

description NETWORK1

ipv6 address 2800:9000:8:1::1/64

ip address 192.168.20.2/24

ipv6 address 2800:9000:8:1::1/64

ipv6 nd suppress-ra

!

interface eth1

description NETWORK2

ip address 192.168.30.2/24

ipv6 address 2800:9000:8:2::1/64

ipv6 nd suppress-ra

!

interface lo

!

interface sit0

ipv6 nd suppress-ra

!

ipv6 forwarding

!

line vty

!

Con el comando show running-config, se visualiza la configuración acutal, en este informe debe de verificarse que se encuentra activado el forward para ipv6, a través del comando ***ipv6 forwarding****, si éste no aparece en el resultado del comando show running-config, entonces debe de realizar los pasos de su activación, como fueron detallados en la sección 11.3.3.2 "* ***Comprobando las herramientas de red para IPv6****"*

b. show ipv6 route: tiene la misma función que show ip route que se utiliza para el protocolo ipv4, con la diferencia, que este comando presenta las redes de ipv6, que se encuentran conectadas directamente a una interfaz, o que han sido configuradas previamente por medio de protocolos de enrutamiento. En el siguiente resultado se presentan las redes que pertenecen al link local y las globales.

show ipv6 route

Root# show ipv6 route

Codes: K - kernel route, C - connected, S - static, R - RIPng, O - OSPFv3,

I - ISIS, B - BGP, * - FIB route.

C>* ::1/128 is directly connected, lo

K * 2800:9000:8:1::/64 is directly connected, eth0

C>* 2800:9000:8:1::/64 is directly connected, eth0

K * 2800:9000:8:2::/64 is directly connected, eth1

C>* 2800:9000:8:2::/64 is directly connected, eth1

K>* fe80::/64 is directly connected, eth0

K>* fe80::/64 is directly connected, eth1

K>* ff00::/8 is directly connected, eth0

K>* ff00::/8 is directly connected, eth1

En el resultado del comando show ipv6 route, se visualizan las direcciones de cada red configurada para cada interfaz, así como también, se visualizan las direcciones de red de tipo enlace local, establecidas automáticamente. Si no aparecen las direcciones de tipo de enlace local (fe80::/64) entonces, se debe de revisar cuidadosamente si el módulo de ipv6 está cargado correctamente.

b. show ipv6 forwarding: con este comando se verifica si el forward de ipv6, es decir, el reenvío de paquetes para ipv6 se encuentra activado. Si aparece en off, significa que no está activado y es necesario activarlo.

show ipv6 forwarding

Root# show ipv6 forwarding

ipv6 forwarding is on

También se pueden utilizar comandos para verificar la configuración del protocolo ipv4.

a. show ip route: muestra las redes que se encuentran directamente conectadas, o las que han sido configuradas por protocolos de enrutamiento como RIP, OSPF, etc.

show ip route

Root# show ip route

Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,

I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo

K>* 169.254.0.0/16 is directly connected, lo

C>* 192.168.20.0/24 is directly connected, eth0

C>* 192.168.30.0/24 is directly connected, eth1

b. show ip forwarding: con este comando se verifica si el forward de ip, es decir, el reenvío de paquetes para ip se encuentra activado. Si aparece en off, sifignifica que no está activado y es necesario activarlo.

show ip forwarding

Root# show ip forwarding

IP forwarding is on

Si el resultado es "off", entonces se debe de activiar el forward, desde el terminal: Echo 1 > /proc/sys/net/ipv4/conf/all/forwarding

Para comprobar que las direcciones ipv6 fueron configuradas correctamente, se procede a ejecutar ping6 a cada dirección.
Dentro de Quagga, se coloca de la siguiente forma: **ping ipv6 direccion_ipv6**

ping ipv6 2800:9000:8:1::1

Root# ping ipv6 2800:9000:8:1::1

PING 2800:9000:8:1::1(2800:9000:8:1::1) 56 data bytes

64 bytes from 2800:9000:8:1::1: icmp_seq=0 ttl=64 time=0.074 ms

64 bytes from 2800:9000:8:1::1: icmp_seq=1 ttl=64 time=0.070 ms

64 bytes from 2800:9000:8:1::1: icmp_seq=2 ttl=64 time=0.072 ms

ping ipv6 2800:9000:8:2::1

Root# ping ipv6 2800:9000:8:2::1

PING 2800:9000:8:2::1(2800:9000:8:2::1) 56 data bytes

64 bytes from 2800:9000:8:2::1: icmp_seq=0 ttl=64 time=0.073 ms

64 bytes from 2800:9000:8:2::1: icmp_seq=1 ttl=64 time=0.073 ms

64 bytes from 2800:9000:8:2::1: icmp_seq=2 ttl=64 time=0.072 ms

10.5.2.3 Configuración de Linux, desde la consola.

También se puede manejar el protocolo de IPv6 desde el Terminal de Linux, a continuación se presentan algunos comandos útiles:

1. Mostrar direcciones IPv6
/sbin/ip -6 addr show dev <interface>
/sbin/ifconfig <interface>

2 Añadir una dirección IPv6. Se puede hacer mediante el uso de **ip** o **ipconfig:**
/sbin/ip -6 addr add <ipv6address>/<prefixlength> dev <interface>
/sbin/ifconfig <interface> inet6 add <ipv6address>/<prefixlength>

3. Eliminar una dirección IPv6

/sbin/ip -6 addr del <ipv6address>/<prefixlength> dev <interface>

/sbin/ifconfig <interface> inet6 del <ipv6address>/<prefixlength>

4. Mostrar rutas IPv6. Se puede hacer mediante el uso de **ip** o **route.**

/sbin/ip -6 route show [dev <device>]

/sbin/route –A inet6

5. Añadir una ruta IPv6 a través de un gateway. Se puede hacer mediante el uso de **ip** o **route**.

/sbin/ip -6 route add <ipv6network>/<prefixlength> via <ipv6address> [dev <device>]

/sbin/route –A inet6 add <ipv6network>/<prefixlength> gw <ipv6address> [dev <device>]

6. Eliminar una ruta IPv6 a través de un gateway. Se puede hacer mediante el uso de ip o route:

/sbin/ip -6 route del <ipv6network>/<prefixlength> via <ipv6address> [dev <device>]

/sbin/route –A inet6 del <ipv6network>/<prefixlength> via <ipv6address> [dev <device>]

7. Añadir una ruta IPv6 a través de una interfaz

/sbin/ip -6 route add <ipv6network>/<prefixlength> dev <device>

/sbin/route –A inet6 add <ipv6network>/<prefixlength> dev <device>

8. Eliminar una ruta IPv6 a través de una interfaz

/sbin/ip -6 route del <ipv6network>/<prefixlength> dev <device>

/sbin/route –A inet6 del <ipv6network>/<prefixlength> dev <device>

Ahora desde la consola se procede a verificar que las direcciones ingresadas a través de Quagga estén correctas, para ello, se emplea el comando ifconfig, que muestra toda la información relacionada a las interfaces, como la dirección MAC, dirección de enlace local, dirección global (asignada manualmente) etc. Otro comando útil a emplear es el ping6 que se utiliza de manera distinta en direcciones de enlace local y global.

Empleando el comando **ifconfig**:

```
[root@Root ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:76:EE:FE:A7
          inet6 addr: 2800:9000:8:1::1/64 Scope:Global
          inet6 addr: fe80::20c:76ff:feee:fea7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1979 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1036 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:228688 (223.3 KiB)  TX bytes:121756 (118.9 KiB)
          Interrupt:5

eth1      Link encap:Ethernet  HWaddr 00:08:A1:A5:4C:C8
          inet6 addr: 2800:9000:8:2::1/64 Scope:Global
          inet6 addr: fe80::208:a1ff:fea5:4cc8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:956 errors:0 dropped:0 overruns:0 frame:0
          TX packets:385 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:118580 (115.8 KiB)  TX bytes:43310 (42.2 KiB)
          Interrupt:10 Base address:0xac00

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:2895 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2895 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4581176 (4.3 MiB)  TX bytes:4581176 (4.3 MiB)
```

Para realizar ping6 a una dirección de enlace local, se debe de hacer de la siguiente manera: **ping6 –I interfaz dirección_local.** Así como se muestra a continuación:

```
ping6 -I eth0 fe80::20c:76ff:feee:fea7
PING fe80::20c:76ff:feee:fea7(fe80::20c:76ff:feee:fea7) from ::1 eth0: 56 data bytes
64 bytes from fe80::20c:76ff:feee:fea7: icmp_seq=0 ttl=64 time=0.085 ms
64 bytes from fe80::20c:76ff:feee:fea7: icmp_seq=1 ttl=64 time=0.086 ms
64 bytes from fe80::20c:76ff:feee:fea7: icmp_seq=2 ttl=64 time=0.086 ms
64 bytes from fe80::20c:76ff:feee:fea7: icmp_seq=3 ttl=64 time=0.085 ms
```

```
[root@Root ~]# ping6 -I eth1 fe80::208:a1ff:fea5:4cc8
PING fe80::208:a1ff:fea5:4cc8(fe80::208:a1ff:fea5:4cc8) from ::1 eth1: 56 data bytes
64 bytes from fe80::208:a1ff:fea5:4cc8: icmp_seq=0 ttl=64 time=0.128 ms
64 bytes from fe80::208:a1ff:fea5:4cc8: icmp_seq=1 ttl=64 time=0.086 ms
64 bytes from fe80::208:a1ff:fea5:4cc8: icmp_seq=2 ttl=64 time=0.084 ms
```

Las direcciones de tipo enlace local, no se pueden verificar a través de zebra, solamente por medio de la consola.

Para las direcciones globales, se utiliza el comando ping6 de la misma forma que el ping para ipv4.

```
[root@Root ~]# ping6 2800:9000:8:1::1
PING 2800:9000:8:1::1(2800:9000:8:1::1) 56 data bytes
64 bytes from 2800:9000:8:1::1: icmp_seq=0 ttl=64 time=0.076 ms
64 bytes from 2800:9000:8:1::1: icmp_seq=1 ttl=64 time=0.080 ms
64 bytes from 2800:9000:8:1::1: icmp_seq=2 ttl=64 time=0.080 ms
64 bytes from 2800:9000:8:1::1: icmp_seq=3 ttl=64 time=0.077 ms
```

```
 [root@Root ~]# ping6 2800:9000:8:2::1
PING 2800:9000:8:2::1(2800:9000:8:2::1) 56 data bytes
64 bytes from 2800:9000:8:2::1: icmp_seq=0 ttl=64 time=0.082 ms
64 bytes from 2800:9000:8:2::1: icmp_seq=1 ttl=64 time=0.080 ms
64 bytes from 2800:9000:8:2::1: icmp_seq=2 ttl=64 time=0.081 ms
```

El comando **ip -6 route** funciona de la misma manera que show ipv6 route, y muestra la misma información también.

Por medio de este, se verifican las redes que están asociadas a cada interfaz. Puede ver el siguiente ejemplo:

```
[root@Root ~]# ip -6 route
2800:9000:8:1::/64 dev eth0  metric 256  mtu 1500 advmss 1440 metric10 64
2800:9000:8:2::/64 dev eth1  metric 256  mtu 1500 advmss 1440 metric10 64
fe80::/64 dev eth0  metric 256  mtu 1500 advmss 1440 metric10 64
fe80::/64 dev eth1  metric 256  mtu 1500 advmss 1440 metric10 64
ff00::/8 dev eth0  metric 256  mtu 1500 advmss 1440 metric10 1
ff00::/8 dev eth1  metric 256  mtu 1500 advmss 1440 metric10 1
```

10.5.2.4 Activando el servicio de ip6tables.

En este documento existe una sección que trata sobre la aplicación de ip6tables para un firewall, anteriormente en el capítulo 8 se explicó que es un firewall. En esta sección no se utiliza ip6tables para denegar el acceso a ciertas computadoras, si no que, se utiliza para permitir el libre tráfico entre las dos interfaces.

Al igual que el forwarding, la verificación del estado de ip6tables es necesaria para permitir la comunicación entre dos subredes diferentes conectadas en cada interfaz. Si el forwarding aparece como desactivado, y el estado de ip6tables es eliminar el tráfico, entonces se tendría dos subredes aisladas, es decir cada host de cada sub red podría hacer ping entre ellas y solamente llegar a la interfaz correspondiente de la sub red y nunca pasaría a la otra interfaz.

1. El primer paso que se debe realizar es desactivar el iptables, que funciona para ipv4. La instrucción que se utiliza es **service iptables stop.**

```
[root@Root ~]# service iptables stop
Expurgar reglas del cortafuegos:                           [  OK  ]
Configuración de cadenas a la política ACCEPT: filter      [  OK  ]
Descargando módulos iptables:                              [  OK  ]
```

2. Para mayor seguridad, antes de activar el servicio de ip6tables se verifica si en realidad el módulo de ip6tables se encuentra cargado:

```
[root@Root ~]# lsmod | grep -w 'ip6_tables' && echo "modulo cargado"
ip6_tables             17729  1 ip6table_filter
modulo cargado
```

3. Después que se verifica que el módulo esta cargado correctamente, se procede a iniciar el servicio, por medio de **service ip6tables Start.**

```
root@Root ~]# service ip6tables start
Aplicando reglas del cortafuegos ip6tables:                [  OK  ]
```

Para que el módulo se cargue automáticamente cada vez que se inicie la sesión se utiliza la siguiente instrucción

```
[root@Root ~]# chkconfig --level 345 ip6tables on
```

Otra forma de iniciar el servicio y guardarlo para que inicie siempre correctamente, es a través del menú de aplicaciones, seleccione configuración del sistema, y escoja configuración de servidores y dé un clic en servicios. Seleccione ip6tables posteriormente de un clic en el botón <**inicio**> y luego un clic en el botón <**guardar**>

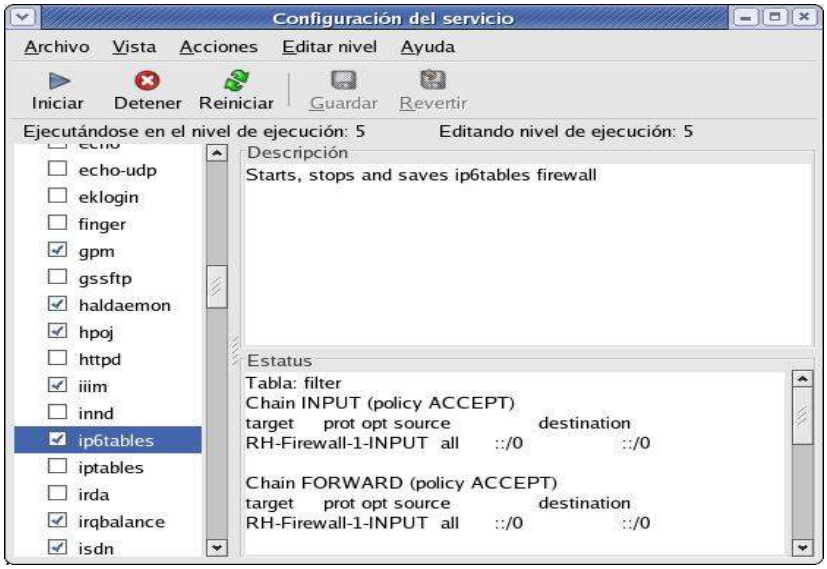


Figura 77. Iniciando y guardando el servicio de ip6tables.

3. Una vez iniciado el servicio, en la consola se digita lo siguiente:

```
[root@Root ~]# ip6tables -F
[root@Root ~]# ip6tables -X
[root@Root ~]# ip6tables -Z
```

Las instrucciones anteriores se utilizan para borrar cualquier cadena previamente ingresada.

Ip6tables -F: borrar todas las reglas una por una, si existen.

Ip6tables -X: borra las opciones de los usuarios definidos en la cadena especificada.

Ip6tables -Z: activa el Zero y los contadores byte en todas las cadenas.

4. Luego se procede a guardar el archivo de ip6tables, con la siguiente instrucción:

```
[root@Root ~]# service ip6tables save
[  OK  ]
```

El archivo se almacena en el directorio **/etc/sysconfig/**

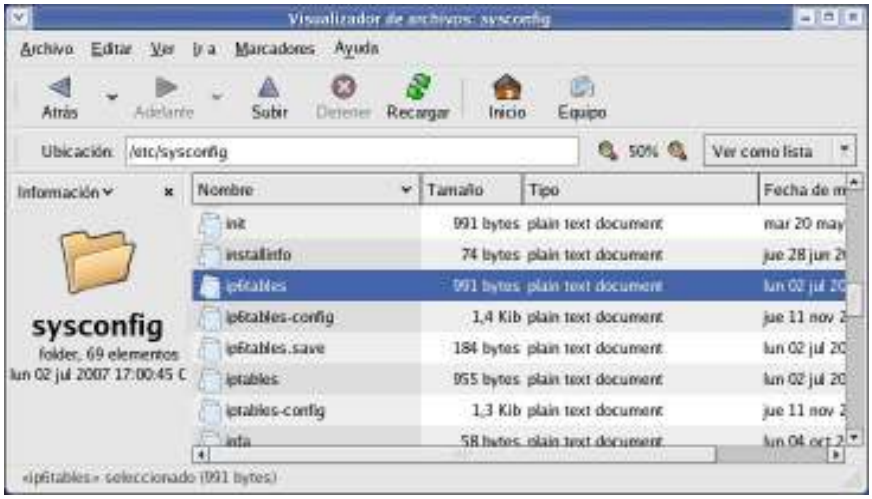


Figura 78. Las reglas se almacenan en el archivo ip6tables.

5. Se inicia de nuevo el servicio, o también se puede utilizar el comando service ip6tables restart.

```
[root@Root ~]# service ip6tables start
Expurgar reglas del cortafuegos:                    [  OK  ]
Configuración de cadenas a la política ACCEPT: filter      [  OK  ]
Descargando módulos ip6tables:                      [  OK  ]
```

6. Se utiliza la instrucción ip6tables –L, para ver las cadenas almacenadas en el archivo ip6tables. En este caso, la entrada, el reenvío y salida de paquetes está permitida.

```
[root@Root ~]# ip6tables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Por defecto, el archivo **/etc/sysconfig/ip6tables**, cuando se activa, tiene el siguiente contenido:

Contenido del archive /etc/sysconfig/ip6tables

```
ip6tables

# Generated by ip6tables-save v1.2.11 on Thu Jul  5 11:09:16 2007

*filter

:INPUT ACCEPT [0:0]

:FORWARD ACCEPT [0:0]

:OUTPUT ACCEPT [0:0]

:RH-Firewall-1-INPUT - [0:0]

-A INPUT -j RH-Firewall-1-INPUT

-A FORWARD -j RH-Firewall-1-INPUT

-A RH-Firewall-1-INPUT -i lo -j ACCEPT
```

-A RH-Firewall-1-INPUT -i eth0 -j ACCEPT

-A RH-Firewall-1-INPUT -i eth1 -j ACCEPT

-A RH-Firewall-1-INPUT -p icmpv6 -j ACCEPT

-A RH-Firewall-1-INPUT -p 50 -j ACCEPT

-A RH-Firewall-1-INPUT -p 51 -j ACCEPT

-A RH-Firewall-1-INPUT -p udp --dport 5353 -d ff02::fb -j ACCEPT

-A RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT

-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 631 -j ACCEPT

COMMIT

10.6 Comprobación de la conectividad

Una vez se realiza la conectividad básica de ipv6 en cada uno de los computadoras, se procede a verificar la conectividad entre cada una de ellas. Es decir, las computadoras que pertenecen a una subred si son capaces de realizar ping6 a las que pertenecen a otra subred.

10.6.1 En plataforma Windows XP

Además de hacer ping6 a los hosts vecinos, también se empleará los comandos mencionados en la sección 11.5 “Configuración básica en Windows XP”:

Ipv6 rt: se utiliza para visualizar las rutas existentes

Ipv6 rc: se utiliza para ver la entrada del caché de enrutamiento

Ipv6 nc: se utiliza para visualizar a los vecinos, los clasifica por identificador de interfaz.

Netsh interface ipv6 show address: se utiliza para ver todas las direcciones asignadas y configuradas para cada interfaz.

Netsh interface ipv6 show routes: su funcionalidad es verificar las rutas existentes de una menera mas detallada que el comando rt.

Netsh interface ipv6 show neighbors: muestra a los vecinos de una forma más detallada que el comando ipv6 nc.

Conectividad en PC1

Ping6 a la PC 2 que pertenece a la misma subred:

```
D:\>ping6 2800:9000:8:2::3

Haciendo ping 2800:9000:8:2::3
de 2800:9000:8:2::2 con 32 bytes de datos:

Respuesta desde 2800:9000:8:2::3: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::3: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::3: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::3: bytes=32 tiempo<1m

Estadísticas de ping para 2800:9000:8:2::3:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Ping6 al gateway que es la eth1 del router.

```
D:\>ping6 2800:9000:8:2::1

Haciendo ping 2800:9000:8:2::1
de 2800:9000:8:2::2 con 32 bytes de datos:

Respuesta desde 2800:9000:8:2::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::1: bytes=32 tiempo<1m

Estadísticas de ping para 2800:9000:8:2::1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Ping6 a la eth0 del router que pertenece a la subred 2800:9000:8:1::/48

```
D:\>ping6 2800:9000:8:1::1

Haciendo ping 2800:9000:8:1::1
de 2800:9000:8:2::2 con 32 bytes de datos:

Respuesta desde 2800:9000:8:1::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::1: bytes=32 tiempo<1m

Estadísticas de ping para 2800:9000:8:1::1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Ping6 a la Pc3 que pertenece a la subred 2800:9000:8:1::/48

```
D:\>ping6 2800:9000:8:1::2

Haciendo ping 2800:9000:8:1::2
de 2800:9000:8:2::2 con 32 bytes de datos:

Respuesta desde 2800:9000:8:1::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::2: bytes=32 tiempo<1m

Estadísticas de ping para 2800:9000:8:1::2:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Ping6 a la Pc4 que pertenece a la subred 2800:9000:8:1::/48

```
D:\>ping6 2800:9000:8:1::3

Haciendo ping 2800:9000:8:1::3
de 2800:9000:8:2::2 con 32 bytes de datos:

Respuesta desde 2800:9000:8:1::3: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::3: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::3: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::3: bytes=32 tiempo<1m

Estadísticas de ping para 2800:9000:8:1::3:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Verificando la ruta por defecto, en este caso es la ruta que lleva hacia a la puerta de enlace.

```
D:\>ipv6 rt
::/0 -> 4/2800:9000:8:2::1 pref 0 duración infinite (manual)
```

Verificando la entrada de la caché de enrutamiento.

```
D:\>ipv6 rc
2800:9000:8:2::3 vía 4/2800:9000:8:2::1
    src 4/2800:9000:8:2::2
    PMTU 1500
2800:9000:8:1::3 vía 4/2800:9000:8:2::1
    src 4/2800:9000:8:2::2
    PMTU 1500
2800:9000:8:1::2 vía 4/2800:9000:8:2::1
    src 4/2800:9000:8:2::2
    PMTU 1500
2800:9000:8:2::1 vía 4/2800:9000:8:2::1
```

```
    src 4/2800:9000:8:2::2
    PMTU 1500
2800:9000:8:1::1 vía 4/2800:9000:8:2::1
    src 4/2800:9000:8:2::2
    PMTU 1500
2800:9000:8:2::2 vía 4/2800:9000:8:2::2
    src 4/2800:9000:8:2::2
    PMTU 1500
fe80::2c0:9fff:fed2:b2fd vía 4/fe80::2c0:9fff:fed2:b2fd
    src 4/fe80::2c0:9fff:fed2:b2fd
    PMTU 1500
```

Verificando a los vecinos, clasificados por cada interfaz. Detecta automáticamente al enrutador.

```
D:\>ipv6 nc
5: fe80::5445:5245:444f 0.0.0.0:0        permanente
4:   2800:9000:8:2::1 00-08-a1-a5-4c-c8 accesible (8000ms) (enrutador)
4:   2800:9000:8:2::2 00-c0-9f-d2-b2-fd permanente
4: fe80::2c0:9fff:fed2:b2fd 00-c0-9f-d2-b2-fd permanente
4: fe80::208:a1ff:fea5:4cc8 00-08-a1-a5-4c-c8 obsoleto
4:   2800:9000:8:2::3 00-15-60-af-32-32 obsoleto
2: fe80::5efe:192.168.30.3 127.0.0.1        permanente
1:          fe80::1                permanente
1:             ::1                 permanente
```

Verificando las direcciones que han sido configuradas en cada interfaz.

```
D:\>netsh interface ipv6 show address
Consultando el estado activo...

Interfaz 5: Teredo Tunneling Pseudo-Interface

Tipo dir.  Estado DAD Vida válida  Vida pref.   Dirección
---------  ---------- ------------ ------------ ----------------------------
Vínculo    Preferida      infinite     infinite fe80::5445:5245:444f

Interfaz 4: Conexión de área local

Tipo dir.  Estado DAD Vida válida  Vida pref.   Dirección
---------  ---------- ------------ ------------ ----------------------------
Manual     Preferida      infinite     infinite 2800:9000:8:2::2
Vínculo    Preferida      infinite     infinite fe80::2c0:9fff:fed2:b2fd

Interfaz 2: Automatic Tunneling Pseudo-Interface

Tipo dir.  Estado DAD Vida válida  Vida pref.   Dirección
---------  ---------- ------------ ------------ ----------------------------
Vínculo    Preferida      infinite     infinite fe80::5efe:192.168.30.3

Interfaz 1: Loopback Pseudo-Interface

Tipo dir.  Estado DAD Vida válida  Vida pref.   Dirección
```

```
---------  ---------- ------------ ------------ ----------------------------
Bucle inv. Preferida      infinite     infinite ::1
Vínculo    Preferida      infinite     infinite fe80::1
```

El siguiente comando se utiliza para verificar la ruta que lleva a la puerta de enlace.

```
D:\>netsh interface ipv6 show routes
Consultando el estado activo...

Publicar Tipo       Mét  Prefijo                Índ  Interfaz/puerta_enlace
-------- -------- ---- ---------------------- ---  ----------------------
no       Manual     0  ::/0                    4  2800:9000:8:2::1
```

El siguiente comando se utiliza para visualizar a los vecinos de una manera detallada por cada interfaz:

```
D:\>netsh interface ipv6 show neighbors

Interfaz 5: Teredo Tunneling Pseudo-Interface

Dirección de Internet                   Dirección física   Tipo
--------------------------------------  -----------------  -----------
fe80::5445:5245:444f                      0.0.0.0:0          Permanentes

Interfaz 4: Conexión de área local

Dirección de Internet                   Dirección física   Tipo
--------------------------------------  -----------------  -----------
2800:9000:8:2::1                        00-08-a1-a5-4c-c8  Obsoleto (enrutador)
2800:9000:8:2::2                        00-c0-9f-d2-b2-fd  Permanentes
fe80::2c0:9fff:fed2:b2fd                00-c0-9f-d2-b2-fd  Permanentes
fe80::208:a1ff:fea5:4cc8                00-08-a1-a5-4c-c8  Obsoleto
2800:9000:8:2::3                        00-15-60-af-32-32  Obsoleto

Interfaz 2: Automatic Tunneling Pseudo-Interface

Dirección de Internet                   Dirección física   Tipo
--------------------------------------  -----------------  -----------
fe80::5efe:192.168.30.3                   127.0.0.1          Permanentes

Interfaz 1: Loopback Pseudo-Interface

Dirección de Internet                   Dirección física   Tipo
--------------------------------------  -----------------  -----------
fe80::1                                                    Permanentes
::1                                                        Permanentes
```

Conectividad en PC2

Ping6 a la PC 1 que pertenece a la misma subred:

```
C:\>ping6 2800:9000:8:2::2

Haciendo ping 2800:9000:8:2::2
de 2800:9000:8:2::3 con 32 bytes de datos:

Respuesta desde 2800:9000:8:2::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::2: bytes=32 tiempo<1m

Estadísticas de ping para 2800:9000:8:2::2:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Ping6 al gateway que es la eth1 del router.

```
C:\>ping6 2800:9000:8:2::1

Haciendo ping 2800:9000:8:2::1
de 2800:9000:8:2::3 con 32 bytes de datos:

Respuesta desde 2800:9000:8:2::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::1: bytes=32 tiempo<1m

Estadísticas de ping para 2800:9000:8:2::1:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Ping6 a la eth0 del router que pertenece a la subred 2800:9000:8:1::/48

```
C:\>ping6 2800:9000:8:1::1

Haciendo ping 2800:9000:8:1::1
de 2800:9000:8:2::3 con 32 bytes de datos:

Respuesta desde 2800:9000:8:1::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::1: bytes=32 tiempo<1m

Estadísticas de ping para 2800:9000:8:1::1:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Ping6 a la Pc3 que pertenece a la subred 2800:9000:8:1::/48

```
C:\>ping6 2800:9000:8:1::2

Haciendo ping 2800:9000:8:1::2
de 2800:9000:8:2::3 con 32 bytes de datos:

Respuesta desde 2800:9000:8:1::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::2: bytes=32 tiempo<1m

Estadísticas de ping para 2800:9000:8:1::2:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Ping6 a la Pc4 que pertenece a la subred 2800:9000:8:1::/48

```
C:\>ping6 2800:9000:8:1::3

Haciendo ping 2800:9000:8:1::3
de 2800:9000:8:2::3 con 32 bytes de datos:

Respuesta desde 2800:9000:8:1::3: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::3: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::3: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::3: bytes=32 tiempo<1m

Estadísticas de ping para 2800:9000:8:1::3:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Verificando la ruta por defecto, en este caso es la ruta que lleva hacia a la puerta de enlace.

```
C:\>ipv6 rt
::/0 -> 4/2800:9000:8:2::1 pref 0 duración infinite (manual)
```

Verificando la entrada de la caché de enrutamiento

```
C:\>ipv6 rc
2800:9000:8:1::3 vía 4/2800:9000:8:2::1
    src 4/2800:9000:8:2::3
    PMTU 1500
2800:9000:8:1::2 vía 4/2800:9000:8:2::1
    src 4/2800:9000:8:2::3
    PMTU 1500
2800:9000:8:2::1 vía 4/2800:9000:8:2::1
    src 4/2800:9000:8:2::3
    PMTU 1500
2800:9000:8:2::2 vía 4/2800:9000:8:2::1
    src 4/2800:9000:8:2::3
    PMTU 1500
2800:9000:8:1::1 vía 4/2800:9000:8:2::1
    src 4/2800:9000:8:2::3
    PMTU 1500
2800:9000:8:2::3 vía 4/2800:9000:8:2::3
    src 4/2800:9000:8:2::3
    PMTU 1500
fe80::215:60ff:feaf:3232 vía 4/fe80::215:60ff:feaf:3232
    src 4/fe80::215:60ff:feaf:3232
    PMTU 1500
```

Verificando a los vecinos, clasificados por cada interfaz. Detecta automáticamente al enrutador.

```
C:\>ipv6 nc
5: fe80::5445:5245:444f 0.0.0.0:0        permanente
4:   2800:9000:8:2::1 00-08-a1-a5-4c-c8 obsoleto (enrutador)
4:   2800:9000:8:2::3 00-15-60-af-32-32 permanente
4: fe80::215:60ff:feaf:3232 00-15-60-af-32-32 permanente
4: fe80::208:a1ff:fea5:4cc8 00-08-a1-a5-4c-c8 obsoleto
4:   2800:9000:8:2::2                incompleto
2: fe80::5efe:192.168.30.4 127.0.0.1        permanente
1:          fe80::1                  permanente
1:             ::1                permanente
```

Verificando las direcciones ipv6 asignadas para cada interfaz.

```
C:\>netsh interface ipv6 show address
Consultando el estado activo...


Interfaz 5: Teredo Tunneling Pseudo-Interface

Tipo dir.  Estado DAD Vida válida  Vida pref.   Dirección
---------  ---------- ------------ ------------ ----------------------------
Vínculo    Preferida      infinite     infinite fe80::5445:5245:444f

Interfaz 4: Conexión de área local

Tipo dir.  Estado DAD Vida válida  Vida pref.   Dirección
---------  ---------- ------------ ------------ ----------------------------
Manual     Preferida      infinite     infinite 2800:9000:8:2::3
Vínculo    Preferida      infinite     infinite fe80::215:60ff:feaf:3232

Interfaz 2: Automatic Tunneling Pseudo-Interface

Tipo dir.  Estado DAD Vida válida  Vida pref.   Dirección
---------  ---------- ------------ ------------ ----------------------------
Vínculo    Preferida      infinite     infinite fe80::5efe:192.168.30.4

Interfaz 1: Loopback Pseudo-Interface

Tipo dir.  Estado DAD Vida válida  Vida pref.   Dirección
---------  ---------- ------------ ------------ ----------------------------
Bucle inv. Preferida      infinite     infinite ::1
Vínculo    Preferida      infinite     infinite fe80::1
```

El siguiente comando se utiliza para verificar la ruta que lleva a la puerta de enlace.

```
C:\>netsh interface ipv6 show routes
Consultando el estado activo...

Publicar Tipo       Mét  Prefijo                Índ  Interfaz/puerta_enlace
-------- --------  ----  ------------------------  ---  ----------------------
no       Manual       0  ::/0                     4  2800:9000:8:2::1
```

El siguiente comando se utiliza para visualizar a los vecinos de una manera detallada por cada interfaz:

```
C:\>netsh interface ipv6 show neighbors

Interfaz 5: Teredo Tunneling Pseudo-Interface

Dirección de Internet                   Dirección física   Tipo
--------------------------------------- ----------------- -----------
fe80::5445:5245:444f                     0.0.0.0:0         Permanentes

Interfaz 4: Conexión de área local

Dirección de Internet                   Dirección física   Tipo
--------------------------------------- ----------------- -----------
2800:9000:8:2::1                        00-08-a1-a5-4c-c8  Obsoleto (enru
tador)
2800:9000:8:2::3                        00-15-60-af-32-32  Permanentes
fe80::215:60ff:feaf:3232                00-15-60-af-32-32  Permanentes
fe80::208:a1ff:fea5:4cc8                00-08-a1-a5-4c-c8  Obsoleto
2800:9000:8:2::2                                           Incompleto

Interfaz 2: Automatic Tunneling Pseudo-Interface

Dirección de Internet                   Dirección física   Tipo
--------------------------------------- ----------------- -----------
fe80::5efe:192.168.30.4                  127.0.0.1         Permanentes

Interfaz 1: Loopback Pseudo-Interface

Dirección de Internet                   Dirección física   Tipo
--------------------------------------- ----------------- -----------
fe80::1                                                    Permanentes
::1                                                        Permanentes
```

Conectividad en PC3

Ping6 a la PC 4 que pertenece a la misma subred:

```
C:\>ping6 2800:9000:8:1::3

Haciendo ping 2800:9000:8:1::3
de 2800:9000:8:1::2 con 32 bytes de datos:

Respuesta desde 2800:9000:8:1::3: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::3: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::3: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::3: bytes=32 tiempo<1m

Estadísticas de ping para 2800:9000:8:1::3:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Ping6 al gateway que es la eth0 del router.

```
C:\>ping6 2800:9000:8:1::1

Haciendo ping 2800:9000:8:1::1
de 2800:9000:8:1::2 con 32 bytes de datos:

Respuesta desde 2800:9000:8:1::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:1::1: bytes=32 tiempo<1m

Estadísticas de ping para 2800:9000:8:1::1:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Ping6 a la eth1 del router que pertenece a la subred 2800:9000:8:2::/48

```
C:\>ping6 2800:9000:8:2::1

Haciendo ping 2800:9000:8:2::1
de 2800:9000:8:1::2 con 32 bytes de datos:

Respuesta desde 2800:9000:8:2::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::1: bytes=32 tiempo<1m

Estadísticas de ping para 2800:9000:8:2::1:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Ping6 a la Pc1 que pertenece a la subred 2800:9000:8:2::/48

```
C:\>ping6 2800:9000:8:2::2

Haciendo ping 2800:9000:8:2::2
de 2800:9000:8:1::2 con 32 bytes de datos:

Respuesta desde 2800:9000:8:2::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::2: bytes=32 tiempo<1m

Estadísticas de ping para 2800:9000:8:2::2:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Ping6 a la Pc2 que pertenece a la subred 2800:9000:8:2::/48

```
C:\>ping6 2800:9000:8:2::3

Haciendo ping 2800:9000:8:2::3
de 2800:9000:8:1::2 con 32 bytes de datos:

Respuesta desde 2800:9000:8:2::3: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::3: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::3: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::3: bytes=32 tiempo<1m

Estadísticas de ping para 2800:9000:8:2::3:
   Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
   Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Verificando la ruta por defecto, en este caso es la ruta que lleva hacia a la puerta de enlace.

```
C:\>ipv6 rt
::/0 -> 4/2800:9000:8:1::1 pref 0 duración infinite (manual)
```

Verificando la entrada de la caché de enrutamiento

```
C:\>ipv6 rc
2800:9000:8:1::3 vía 4/2800:9000:8:1::1
    src 4/2800:9000:8:1::2
    PMTU 1500
2800:9000:8:2::3 vía 4/2800:9000:8:1::1
    src 4/2800:9000:8:1::2
    PMTU 1500
2800:9000:8:2::2 vía 4/2800:9000:8:1::1
    src 4/2800:9000:8:1::2
    PMTU 1500
2800:9000:8:2::1 vía 4/2800:9000:8:1::1
    src 4/2800:9000:8:1::2
    PMTU 1500
2800:9000:8:1::2 vía 4/2800:9000:8:1::2
    src 4/2800:9000:8:1::2
    PMTU 1500
fe80::208:c7ff:fe56:bedc vía 4/fe80::208:c7ff:fe56:bedc
    src 4/fe80::208:c7ff:fe56:bedc
    PMTU 1500
2800:9000:8:1::1 vía 4/2800:9000:8:1::1
    src 4/2800:9000:8:1::2
    PMTU 1500
```

Verificando a los vecinos, clasificados por cada interfaz. Detecta automáticamente al enrutador.

```
C:\>ipv6 nc
5: fe80::ffff:ffff:fffd 0.0.0.0:0        permanente
4:   2800:9000:8:1::1 00-0c-76-ee-fe-a7 obsoleto (enrutador)
4:   2800:9000:8:1::2 00-08-c7-56-be-dc permanente
4: fe80::208:c7ff:fe56:bedc 00-08-c7-56-be-dc permanente
4: fe80::20c:76ff:feee:fea7 00-0c-76-ee-fe-a7 obsoleto
2: fe80::5efe:192.168.20.3 127.0.0.1        permanente
1:          fe80::1                 permanente
1:             ::1                permanente
```

Verificando las direcciones ipv6 asignadas para cada interfaz.

```
C:\>netsh interface ipv6 show address
Consultando el estado activo...


Interfaz 5: Teredo Tunneling Pseudo-Interface

Tipo dir.  Estado DAD Vida válida  Vida pref.   Dirección
---------  ---------- ------------ ------------ ----------------------------
Vínculo    Preferida      infinite     infinite fe80::ffff:ffff:fffd

Interfaz 4: Conexión de área local 2

Tipo dir.  Estado DAD Vida válida  Vida pref.   Dirección
```

```
--------- ---------- ----------- ----------- ----------------------------
Manual    Preferida     infinite    infinite 2800:9000:8:1::2
Vínculo   Preferida     infinite    infinite fe80::208:c7ff:fe56:bedc

Interfaz 2: Automatic Tunneling Pseudo-Interface

Tipo dir.  Estado DAD Vida válida  Vida pref.   Dirección
---------  ---------- ----------- ----------- ----------------------------
Vínculo    Preferida     infinite    infinite fe80::5efe:192.168.20.3

Interfaz 1: Loopback Pseudo-Interface

Tipo dir.  Estado DAD Vida válida  Vida pref.   Dirección
---------  ---------- ----------- ----------- ----------------------------
Bucle inv. Preferida     infinite    infinite ::1
Vínculo    Preferida     infinite    infinite fe80::1


C:\>netsh interface ipv6 show routes
Consultando el estado activo...

Publicar Tipo       Mét  Prefijo                Índ  Interfaz/puerta_enlace
-------- -------- ----  ---------------------  ---  ----------------------
no       Manual       0  ::/0                     4  2800:9000:8:1::1
```

El siguiente comando se utiliza para visualizar a los vecinos de una manera detallada por cada interfaz:

```
C:\>netsh interface ipv6 show neighbors

Interfaz 5: Teredo Tunneling Pseudo-Interface

Dirección de Internet                   Dirección física   Tipo
--------------------------------------  -----------------  -----------
fe80::ffff:ffff:fffd                    0.0.0.0:0          Permanentes

Interfaz 4: Conexión de área local 2

Dirección de Internet                   Dirección física   Tipo
--------------------------------------  -----------------  -----------
2800:9000:8:1::1                        00-0c-76-ee-fe-a7  Obsoleto (enru
tador)
2800:9000:8:1::2                        00-08-c7-56-be-dc  Permanentes
fe80::208:c7ff:fe56:bedc                00-08-c7-56-be-dc  Permanentes
fe80::20c:76ff:feee:fea7                00-0c-76-ee-fe-a7  Obsoleto




Interfaz 2: Automatic Tunneling Pseudo-Interface

Dirección de Internet                   Dirección física   Tipo
--------------------------------------  -----------------  -----------
```

```
fe80::5efe:192.168.20.3                127.0.0.1         Permanentes

Interfaz 1: Loopback Pseudo-Interface

Dirección de Internet                  Dirección física   Tipo
---------------------------------------- ----------------  -----------
fe80::1                                                    Permanentes
::1                                                        Permanentes
```

10.6.2 En Windows 2000.

Conectividad en PC4

Ping6 a la PC 3 que pertenece a la misma subred

```
D:\>ping6 2800:9000:8:1::2

Pinging 2800:9000:8:1::2 with 32 bytes of data:

Reply from 2800:9000:8:1::2: bytes=32 time<1ms
Reply from 2800:9000:8:1::2: bytes=32 time<1ms
Reply from 2800:9000:8:1::2: bytes=32 time<1ms
Reply from 2800:9000:8:1::2: bytes=32 time<1ms
```

Ping6 al gateway que es la eth0 del router.

```
D:\>ping6 2800:9000:8:1::1

Pinging 2800:9000:8:1::1 with 32 bytes of data:

Reply from 2800:9000:8:1::1: bytes=32 time<1ms
Reply from 2800:9000:8:1::1: bytes=32 time<1ms
Reply from 2800:9000:8:1::1: bytes=32 time<1ms
Reply from 2800:9000:8:1::1: bytes=32 time<1ms
```

Ping6 a la eth1 del router que pertenece a la subred 2800:9000:8:2::/48

```
D:\>ping6 2800:9000:8:2::1

Pinging 2800:9000:8:2::1 with 32 bytes of data:

Reply from 2800:9000:8:2::1: bytes=32 time<1ms
Reply from 2800:9000:8:2::1: bytes=32 time<1ms
Reply from 2800:9000:8:2::1: bytes=32 time<1ms
Reply from 2800:9000:8:2::1: bytes=32 time<1ms
```

Ping6 a la Pc1 que pertenece a la subred 2800:9000:8:2::/48

```
D:\>ping6 2800:9000:8:2::2

Pinging 2800:9000:8:2::2 with 32 bytes of data:
```

```
Reply from 2800:9000:8:2::2: bytes=32 time<1ms
Reply from 2800:9000:8:2::2: bytes=32 time<1ms
Reply from 2800:9000:8:2::2: bytes=32 time<1ms
Reply from 2800:9000:8:2::2: bytes=32 time<1ms
```

Ping6 a la Pc2 que pertenece a la subred 2800:9000:8:2::/48

```
D:\>ping6 2800:9000:8:2::3

Pinging 2800:9000:8:2::3 with 32 bytes of data:

Reply from 2800:9000:8:2::3: bytes=32 time<1ms
Reply from 2800:9000:8:2::3: bytes=32 time<1ms
Reply from 2800:9000:8:2::3: bytes=32 time<1ms
Reply from 2800:9000:8:2::3: bytes=32 time<1ms
```

Verificando la ruta por defecto, en este caso es la ruta que lleva hacia a la puerta de enlace.

```
D:\>ipv6 rt
::/0 -> 4/2800:9000:8:1::1 pref 0 (lifetime infinite)
```

Verificando la entrada de la caché de enrutamiento

```
D:\>ipv6 rc
2800:9000:8:2::3 via 4/2800:9000:8:1::1
    src 4/2800:9000:8:1::3
    PMTU 1500
2800:9000:8:2::2 via 4/2800:9000:8:1::1
    src 4/2800:9000:8:1::3
    PMTU 1500
2800:9000:8:2::1 via 4/2800:9000:8:1::1
    src 4/2800:9000:8:1::3
    PMTU 1500
2800:9000:8:1::1 via 4/2800:9000:8:1::1
    src 4/2800:9000:8:1::3
    PMTU 1500
2800:9000:8:1::2 via 4/2800:9000:8:1::1
    src 4/2800:9000:8:1::3
    PMTU 1500
2800:9000:8:1::3 via 1/::1
    src 4/2800:9000:8:1::3
    PMTU 1500
fe80::250:fcff:fe77:6ff8 via 1/::1
    src 4/fe80::250:fcff:fe77:6ff8
    PMTU 1500
```

Verificando a los vecinos, clasificados por cada interfaz. Detecta automáticamente al enrutador.

```
D:\>ipv6 nc
4:   2800:9000:8:1::1 00-0c-76-ee-fe-a7 stale (router)
4: fe80::20c:76ff:feee:fea7 00-0c-76-ee-fe-a7 stale
4:   2800:9000:8:1::2                incomplete
3: fe80::250:fcff:fe77:6ff8                incomplete
```

```
1:          ::1          permanent
```

10.6.3 En Linux Red HAT ES 4.

Desde la Pc Linux, que simula a un router, se pueden ver las computadoras qu funcionan como clientes, que se encuentran en cada subred, para ello se emplea el **ping6** y el comando **ip -6 route**, para visualizar las rutas trazadas.

Ping6 a la PC3 que se encuentra en la subred1 2800:9000:8:1::/64

```
[root@Root ~]# ping6 2800:9000:8:1::2
PING 2800:9000:8:1::2(2800:9000:8:1::2) 56 data bytes
64 bytes from 2800:9000:8:1::2: icmp_seq=0 ttl=128 time=0.249 ms
64 bytes from 2800:9000:8:1::2: icmp_seq=1 ttl=128 time=0.229 ms
64 bytes from 2800:9000:8:1::2: icmp_seq=2 ttl=128 time=0.228 ms
64 bytes from 2800:9000:8:1::2: icmp_seq=3 ttl=128 time=0.228 ms
```

Ping6 a la PC4 que se encuentra en la subred1 2800:9000:8:1::/64

```
 [root@Root ~]# ping6 2800:9000:8:1::3
PING 2800:9000:8:1::3(2800:9000:8:1::3) 56 data bytes
64 bytes from 2800:9000:8:1::3: icmp_seq=0 ttl=128 time=0.351 ms
64 bytes from 2800:9000:8:1::3: icmp_seq=1 ttl=128 time=0.278 ms
64 bytes from 2800:9000:8:1::3: icmp_seq=2 ttl=128 time=0.229 ms
64 bytes from 2800:9000:8:1::3: icmp_seq=3 ttl=128 time=0.230 ms
```

Ping6 a la PC1 que se encuentra en la subred2 2800:9000:8:2::/64

```
 [root@Root ~]# ping6 2800:9000:8:2::2
PING 2800:9000:8:2::2(2800:9000:8:2::2) 56 data bytes
64 bytes from 2800:9000:8:2::2: icmp_seq=0 ttl=128 time=0.305 ms
64 bytes from 2800:9000:8:2::2: icmp_seq=1 ttl=128 time=0.231 ms
64 bytes from 2800:9000:8:2::2: icmp_seq=2 ttl=128 time=0.229 ms
```

Ping6 a la PC2 que se encuentra en la subred2 2800:9000:8:2::/64

```
[root@Root ~]# ping6 2800:9000:8:2::3
PING 2800:9000:8:2::3(2800:9000:8:2::3) 56 data bytes
64 bytes from 2800:9000:8:2::3: icmp_seq=0 ttl=128 time=0.256 ms
64 bytes from 2800:9000:8:2::3: icmp_seq=1 ttl=128 time=0.196 ms
64 bytes from 2800:9000:8:2::3: icmp_seq=2 ttl=128 time=0.201 ms
```

Verificando las redes que se encuentran directamente conectadas a las interfaces.

```
[root@Root ~]# ip -6 route
2800:9000:8:1::/64 dev eth0  metric 256  mtu 1500 advmss 1440 metric10 64
2800:9000:8:2::/64 dev eth1  metric 256  mtu 1500 advmss 1440 metric10 64
fe80::/64 dev eth0  metric 256  mtu 1500 advmss 1440 metric10 64
fe80::/64 dev eth1  metric 256  mtu 1500 advmss 1440 metric10 64
ff00::/8 dev eth0  metric 256  mtu 1500 advmss 1440 metric10 1
ff00::/8 dev eth1  metric 256  mtu 1500 advmss 1440 metric10 1
```

10.7 Guía de configuración de servicios HTTP, FTP en Linux

En esta sección se explica como configurar los servicios http y ftp en el sistema operativo Linux. Para el servicio HTTP se utiliza Apache, a partir de la versión 2.0, por que tiene soporte para ipv6, en el caso de los sistemas operativos Windows, a partir del IIS[82] 6.0 se puede trabajar con dirección ipv6, Windows 2003 server, ya tiene incorporado el IIS6.0 y Windows Vista posee la versión IIS 7.0.

10.7.1 Instalación y configuración de Apache 2.2.4

Se puede obtener a partir del siguiente enlace: http://httpd.apache.org/download.cgi.
Luego para instalarlo y configurarlo se siguen los siguientes pasos:

1. Puede copiar el archivo.tar.gz en el directorio /etc/ y luego puede accesar al directorio:

```
# cd /etc/
```

2. Luego se descomprime el archivo

```
# tar xvfz httpd-2.2.4.tar.gz
```

3. Luego se instala el servidor Apache:

```
# cd httpd-2.2.4
# ./configure --enable-so
# make
# make install
```

Los archivos de configuración del apache se almacenan en el directorio **/usr/local/apache2.** Como APACHE 2.2.4 tiene soporte para direcciones ipv6, entonces se debe de configurar el archivo httd.conf, el cual es el fichero principal en donde está la configuración principal de APCHE.

5. Soporte de IPv6

En los sistemas que soportan IPv6 con la librería Apache Portable Runtime, Apache soporta IPv6 listening sockets por defecto. Además, las directivas Listen,

[82] *Es una serie de servicios para los ordenadores que funcionan con Windows. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.*

NameVirtualHost, y VirtualHost soportan direcciones IPv6 numéricas (por ejemplo, "Listen [fe80::1]:8080").
En el archivo httpd.conf se encuentra la línea Listen 80, se puede borrar esa línea o se coloca como comentario y se agrega lo siguiente:
Listen [2800:9000:8:1::2]:80

6. Luego se guarda el archivo y se procede a colocar la página para el servicio, en el directorio **/usr/local/apache2/httpdocs**

7. Para iniciar el servicio, se coloca lo siguiente en el Terminal:

```
# /usr/local/apache2/bin/apachectl start
```

8. Para verificar el funcionamiento correcto del servicio, se ingresa a un navegador, de preferencia FireFox y se ingresa la dirección y el nombre de la página web. Se debe de tener presente la forma de poner las urls en IPv6. La forma es **http://[dir-ipv6]:puerto**. Por ejemplo:
http://[2800:9000:8:1::2]/sistema.html

10.7.2 Instalación y configuración de FTP.

FTP (File Transfer Protocol) o Protocolo de Transferencia de Archivos (o ficheros informáticos) es uno de los protocolos estándar más utilizados en Internet. El servicio utiliza los puertos 20 y 21, exclusivamente sobre TCP. El puerto 20 es utilizado para el flujo de datos entre cliente y servidor. El puerto 21 es utilizando para el envío de órdenes del cliente hacia el servidor. Prácticamente todos los sistemas operativos y plataformas incluyen soporte para FTP, lo que permite que cualquier computadora conectada a una red basada sobre TCP/IP pueda hacer uso de este servicio a través de un cliente FTP.

En Linux se utiliza el demonio **VSFTPD (Very Secure FTP Daemon),** el cual es una herramienta para implementar servidores de archivos a través del protocolo FTP. Se distingue principalmente porque sus valores por defecto son muy seguros y por su sencillez en la configuración.

*En Red Hat ES 4 el demonio ya esta incorporado, solo se necesita configurar e iniciarlo, pero si se tiene una versión anterior, se puede descargar el demonio e instalarlo con **yum -y install vsftpd.***

El archivo de configuración se encuentra en la siguiente direcotorio: **/etc/vsftpd/vsftpd.conf**

El fichero de configuración se llama **vsftpd.conf**, al abrirlo se encuentran todas las opciones del demonio por defecto posee la opcion listen, la cual se utiliza para las direcciones ipv4. Entonces para las direcciones ipv6 se necesita eliminar o poner como comentario esa opción y agregar la siguiente:

Listen_ipv6=yes

Luego se procede a agregar los archivos en el directorio **var/ftp,** y se inicia el servicio ejecutando lo siguiente desde el Terminal:

```
/sbin/service vsftpd start
```

Para verificar el funcionamiento correcto del servicio, se ingresa a FireFox y se coloca la dirección, por ejemplo: **ftp://[2800:9000:8:2::1]**

Figura 79.La dirección ipv6 se encierra entre corchetes.

Los exploradores que soportan las direcciones de ipv6 son FireFox, Konqueror en Linux, también Internet Explorer pero para Windows Server 2000, 2003 y Windows Vista.
Si se posee Windows XP, puede instalar sin ningún problema el navegador FireFox.

10.8 Guía de implementación del Firewall en Linux.

En el capítulo 8 “Firewall”, se explicó los conceptos necesarios y se abordó el tema de netfilter6 e ip6tables, que es para el manejo de ipv6. En esta sección se explica como se utiliza ip6tables para crear un firewall sencillo. Para ello, se necesita de un escenario ejemplo. El cual se detalla a continuación:

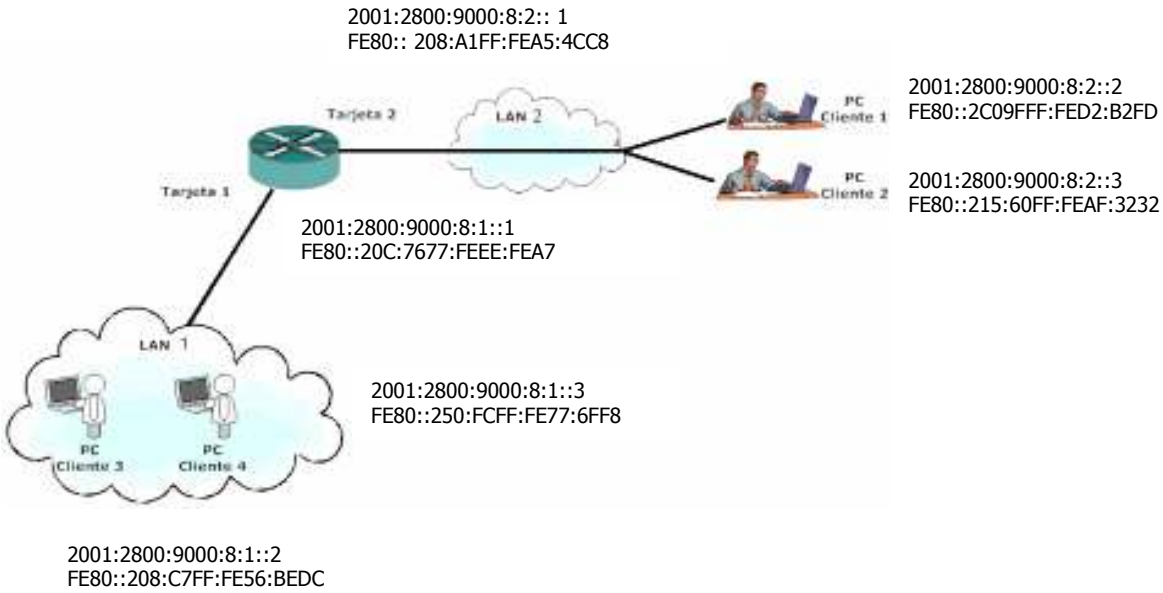


Figura 80. Esquema principal

El esquema anterior, es el esquema principal, donde el router es una máquina Linux. El esquema muestra dos redes separadas, cada una de ellas representa a un departamento dentro de una empresa. En la Lan1 se configura el servicio APACHE, en la Pc Cliente 3, y para ello se le ha configurado Red Hat ES 4. La página que se aloja en apache es un sistema de inventario para ventas de hardware.

En la Lan 2 se configura el servicio APACHE y FTP, pero ninguna de las PC clientes posee el sistema adecuado, por lo tanto se configuran ambos servicios en la interfaz 1, que es la tarjeta2 del Router y que se utiliza como puerta de enlace para la Lan 2. La página de la Lan 2 es sobre la información del proveedor de hardware.
Entonces el esquema se visualiza de a siguiente forma:

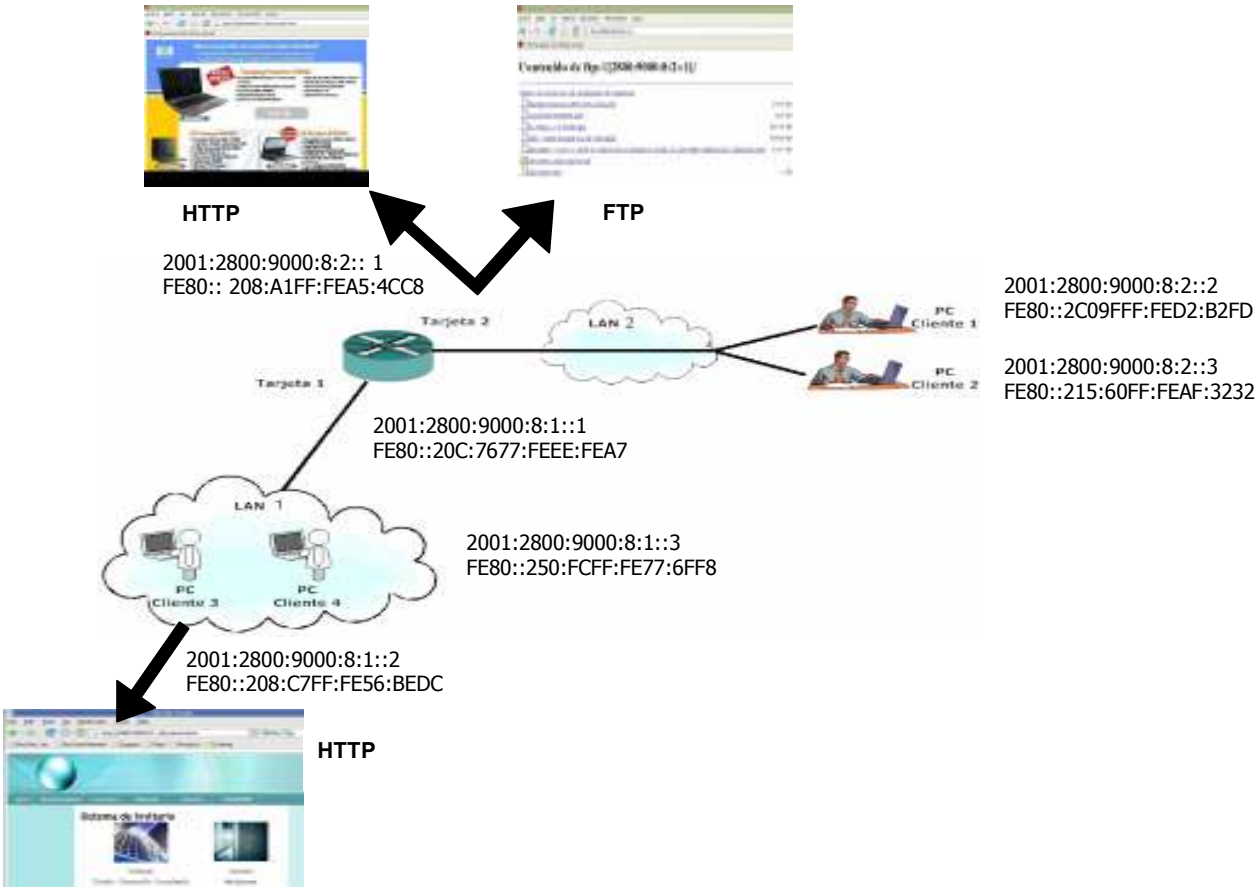


Figura 81. Esquema principal con sus servicios.

De acuerdo a la figura anterior, todas las pc's pueden accesar libremente a los servicios http y ftp, no importando si se encuentran en distintas subredes. Entonces para evitar este problema y separar a los usuarios para que no ingresen a los servicios que no les corresponden, se emplea el Firewall.

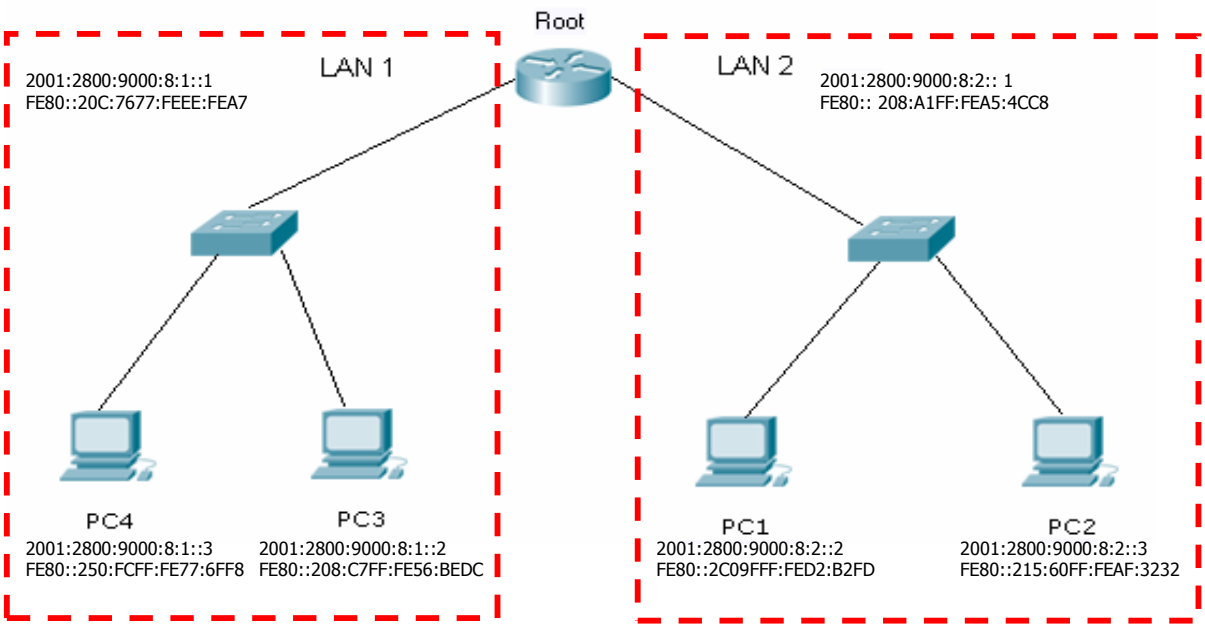


Figura 82. División de las LAN

Las políticas a emplear son las siguientes:

Los usuarios solamente pueden ver a los de la misma red, por lo tanto no podran verse usuarios de distinta red.

- Los Usuarios pc4 y pc3 pueden accesar al sistema de inventarios que se aloja en el servidor apache configurado en la pc3. Y no pueden ver la página de la LAN2.
- Los usuarios pc1 y pc2 pueden ver la página que se aloja en el servidor APACHE, configurado en el Root.
- Solamente la pc2 tiene acceso al servicio FTP configurado en el root.

Pasos a realizar:

1. Se debe de borrar las reglas previamente establecidas con:

```
[root@root ~]# ip6tables –F
# Borra las reglas una por una.
[root@root ~]# ip6tables –X
 Borra las cadenas definidas por los usuarios.
[root@root ~]# ip6tables –Z
# Pone todos los contadores en cero.
```

Para verificar que se borraron las reglas se puede hacer una lista de ellas con el siguiente comando:

```
[root@root ~]# ip6tables –L

# Con –L se hace una lista de las cadenas introducidas por el usuario.

Chain INPUT (policy DROP)
target     prot opt source               destination

# Muestra las cadenas para el parámetro INPUT, en este caso, tiene por defecto DROP, es decir que se rechazan los paquetes que entran en el firewall

Chain FORWARD (policy DROP)
target     prot opt source               destination

# Muestra las cadenas para el parámetro FORWARD, en este caso, tiene por defecto DROP, es decir que se rechaza el reenvío de paquetes en el firewall.

Chain OUTPUT (policy DROP)
target     prot opt source               destination

# Muestra las cadenas para el parámetro INPUT, en este caso, tiene por defecto DROP, es decir que no se permite la salida de los paquetes.
```

Una vez se borran, no se puede hacer ping6, a las otras computadoras, ni a las configuradas en el root.

```
[root@root ~]# ping6 2800:9000:8:2::1

PING 2800:9000:8:2::1(2800:9000:8:2::1) 56 data bytes
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
```

2. Entonces para poder realizar ping6 se coloca como “aceptar” los parámetros INPUT, FORWARD y OUTPUT, a estas se les llama **políticas por defecto**.

```
[root@root ~]# ip6tables –P INPUT  ACCEPT
# se permiten la entrada de los paquetes

[root@root ~]# ip6tables –P FORWARD  ACCEPT
# se permiten el reenvío  de los paquetes

[root@root ~]# ip6tables –P OUTPUT ACCEPT
# se permiten la salida de los paquetes
```

3. Con las políticas por defecto, se permite el libre tránsito de los paquetes entre una lan y la otra, para restringirlo, se emplean las siguientes reglas:

[root@root ~]# **ip6tables -A INPUT -s 2800:9000:8:2::2/128 -d 2800:9000:8:1::1/128 -j DROP**

No se aceptan los paquetes del origen o fuente **(-s)** que provienen de la pc1 (2800:9000:8:2::2/128) que se derigen al destino **(-d)** gateway (2800:9000:8:1::1/128) que pertenece a la LAN1

[root@root ~]# **ip6tables -A INPUT -s 2800:9000:8:2::3/128 -d 2800:9000:8:1::1/128 -j DROP**

No se aceptan los paquetes del origen o fuente **(-s)** que provienen de la pc3 (2800:9000:8:2::3/128) que se derigen al destino **(-d)** gateway (2800:9000:8:1::1/128) que pertenece a la LAN1

[root@root ~]# **ip6tables -A INPUT -s 2800:9000:8:1::2/128 -d 2800:9000:8:2::1/128 -j DROP**

No se aceptan los paquetes del origen o fuente **(-s)** que provienen de la pc2 (2800:9000:8:1::2/128) que se derigen al destino **(-d)** gateway (2800:9000:8:2::1/128) que pertenece a la LAN2

[root@root ~]# **ip6tables -A INPUT -s 2800:9000:8:1::3/128 -d 2800:9000:8:2::1/128 -j DROP**

No se aceptan los paquetes del origen o fuente **(-s)** que provienen de la pc4 (2800:9000:8:1::3/128) que se derigen al destino **(-d)** gateway (2800:9000:8:2::1/128) que pertenece a la LAN2

[root@root ~]# **ip6tables -A FORWARD -s 2800:9000:8:1::/64 -d 2800:9000:8:2::/64 -j DROP**

No se permite el reenvío de paquetes en ambas redes. La red origen o fuente **(-s)** es 2800:9000:8:1::/64 y la red destino **(-d)** es 2800:9000:8:2::/64

[root@root ~]# **ip6tables -A INPUT -s 2800:9000:8:2::2/128 -d 2800:9000:8:2::1/128 -p tcp -m tcp --dport 20:21 -j DROP**

Se restringe el acceso el servicio FTP de la fuente **(-s)** 2800:9000:8:2::2/128 hacia el destino **(-d)** que es la pc1 (2800:9000:8:2::2/128) se especifica el puerto tcp que es el 20:21 para el servicio FTP

La notación 2800:9000:8:1::/64, indica qie es una red, en la cual se ocupan los 64 bits. Y la notación 2800:9000:8:2::3/128 indica solamente la dirección de un host y ocupa los 128. Si se desea colocar una dirección de un host: 2800:/9000:8:1::3/64, entonces netfilter6 toma esta dirección como una red 2800:9000:8:1::, por que solamente se le indicó que tomara 64 bits, por lo tanto no lo toma como un host.

4. Luego se procede a guardarlas reglas:

```
[root@root ~]# service  ip6tables save
# Con esta instrucción se guardan las reglas establecidas anteriormente.
```

5. Una vez guardadas, se procede a verificarlas, las reglas se almacena en el directorio: **/etc/sysconfig/ip6tables** y el archivo contiene lo siguiente:

```
# Generated by ip6tables-save v1.2.11 on Thu Jul 19 15:51:37 2007
*filter

:INPUT ACCEPT [6:416]
:FORWARD ACCEPT [26:2704]
:OUTPUT ACCEPT [2:136]

-A INPUT -s 2800:9000:8:2::2/128 -d 2800:9000:8:1::1/128 -j DROP
-A INPUT -s 2800:9000:8:2::3/128 -d 2800:9000:8:1::1/128 -j DROP
-A INPUT -s 2800:9000:8:1::2/128 -d 2800:9000:8:2::1/128 -j DROP
-A INPUT -s 2800:9000:8:1::3/128 -d 2800:9000:8:2::1/128 -j DROP
-A FORWARD -s 2800:9000:8:1::/64 -d 2800:9000:8:2::/64 -j DROP
-A INPUT -s 2800:9000:8:2::2/128 -d 2800:9000:8:2::1/128  -p tcp -m tcp --dport 20:21 -j DROP
COMMIT
# Completed on Thu Jul 19 15:51:37 2007
```

6. También para verificar la configuración se puede hacer desde la Terminal ingresando el **service ip6tables status.**

```
[root@root ~]# service ip6tables status

# La instrucción service ip6tables status, se utiliza para verificar el estado de las reglas definidas
por el usuario, a continuación se observa la tabla filter, que consta de las cadenas para la entrada
de paquetes (INPUT), reenvío de paquetes (FORWARD) y salida de paquetes (OUTPUT)

Tabla: filter
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all      2800:9000:8:2::2/128  2800:9000:8:1::1/128
DROP       all      2800:9000:8:2::3/128  2800:9000:8:1::1/128
DROP       all      2800:9000:8:1::2/128  2800:9000:8:2::1/128
DROP       all      2800:9000:8:1::3/128  2800:9000:8:2::1/128
DROP       tcp       2800:9000:8:2::2/128  2800:9000:8:2::1/128tcp dpts:20:21

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
DROP       all      2800:9000:8:1::/64   2800:9000:8:2::/64

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

El mismo resultado se obtiene con el comando ip6tables -L

```
[root@root ~]# ip6tables  -L

# Al igual que la instrucción anterior service ip6tables status, muestra por medio de una Lista las
cadenas previamente creadas, a continuación se observan que es lo que se permite o se niega el
acceso en el firewall.

Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all      2800:9000:8:2::2/128  2800:9000:8:1::1/128
DROP       all      2800:9000:8:2::3/128  2800:9000:8:1::1/128
DROP       all      2800:9000:8:1::2/128  2800:9000:8:2::1/128
DROP       all      2800:9000:8:1::3/128  2800:9000:8:2::1/128
DROP       tcp      2800:9000:8:2::2/128  2800:9000:8:2::1/128tcp dpts:ftp-data:ftp

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
DROP       all      2800:9000:8:1::/64   2800:9000:8:2::/64

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Si la información se quiere visualizar de una manera más detallada, entonces se emplea ip6tables –n –v -L

```
[root@root ~]# ip6tables -n -v –L

# La lista que muestra es más detallada por que muestra los paquetes y los bytes que ocupan,
para la entrade de ellos, reenvío y salida)

Chain INPUT (policy ACCEPT 31 packets, 2022 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       all      *      *       2800:9000:8:2::2/128  2800:9000:8:1::1/128
    0     0 DROP       all      *      *       2800:9000:8:2::3/128  2800:9000:8:1::1/128
    0     0 DROP       all      *      *       2800:9000:8:1::2/128  2800:9000:8:2::1/128
    0     0 DROP       all      *      *       2800:9000:8:1::3/128  2800:9000:8:2::1/128
    3   240 DROP       tcp      *      *       2800:9000:8:2::2/128  2800:9000:8:2::1/128tcp dpts:20:21

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       all      *      *       2800:9000:8:1::/64   2800:9000:8:2::/64

Chain OUTPUT (policy ACCEPT 32 packets, 3100 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

6. Para hacer efectivo el cambio en las reglas, se procede a reiniciar el servicio

```
[root@root ~]# service ip6tables restart

# Con esta instrucción se reinicia el servicio de ip6tables
```

7. Para finalizar se verifica las restricciones en los clientes:

Pc2 cliente, puede hacer ping6 a la puerta de enlace y a Pc1

Puerta de enlace:

```
C:\>ping6 2800:9000:8:2::1

Haciendo ping 2800:9000:8:2::1
de 2800:9000:8:2::3 con 32 bytes de datos:

Respuesta desde 2800:9000:8:2::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::1: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::1: bytes=32 tiempo<1m

Estadísticas de ping para 2800:9000:8:2::1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Conectándose con PC1

```
C:\>ping6 2800:9000:8:2::2

Haciendo ping 2800:9000:8:2::2
de 2800:9000:8:2::3 con 32 bytes de datos:
Respuesta desde 2800:9000:8:2::2: bytes=32 tiempo=2ms
Respuesta desde 2800:9000:8:2::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::2: bytes=32 tiempo<1m
Respuesta desde 2800:9000:8:2::2: bytes=32 tiempo<1m
Estadísticas de ping para 2800:9000:8:2::2:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
  Mínimo = 0ms, Máximo = 2ms, Media = 0ms
```

Pero no puede accesar a las otras computadoras de la otra red.

```
C:\>ping6 2800:9000:8:1::1
Haciendo ping 2800:9000:8:1::1
de 2800:9000:8:2::3 con 32 bytes de datos:
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Tiempo de espera agotado para esta solicitud.
Estadísticas de ping para 2800:9000:8:1::1:
    Paquetes: enviados = 4, recibidos = 0, perdidos = 4 (100% perdidos),
```

Y puede accesar a la página de la red y al servicio FTP:

Figura 83. Desde la Pc2 se puede accesar al servicio http y FTP de la subred 2800:9000:8:2::/64

Sin embargo, PC2 no puede ver el servicio http de la subred1 que es la 2800:9000:8:1::/64

.

Figura 84. Desde la Pc2 no logra conectarse con el sistema que se encuentra alojado en pc3 2800:9000:8:2::/64

En la siguiente figura se observa que pc1 tiene restringido el acceso al servicio FTP, aunque pertenezca a la misma red de pc2.

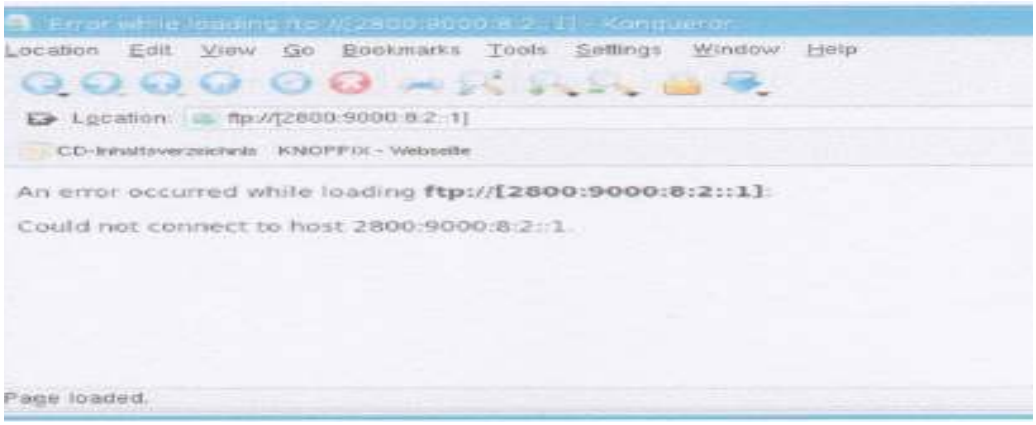


Figura 85. Desde la Pc2 no logra conectarse con el sistema que se encuentra alojado en pc3 2800:9000:8:2::/64

En la otra red que es la 2800:9000:8:1::1/64, la pc3 y pc4 pueden ingresar al sistema con facilidad, pero no pueden establecer conexión con los servicios http ni FTP de la otra red.

Figura 86. Desde la Pc3 y PC4 pueden ingresar al sistema de inventario

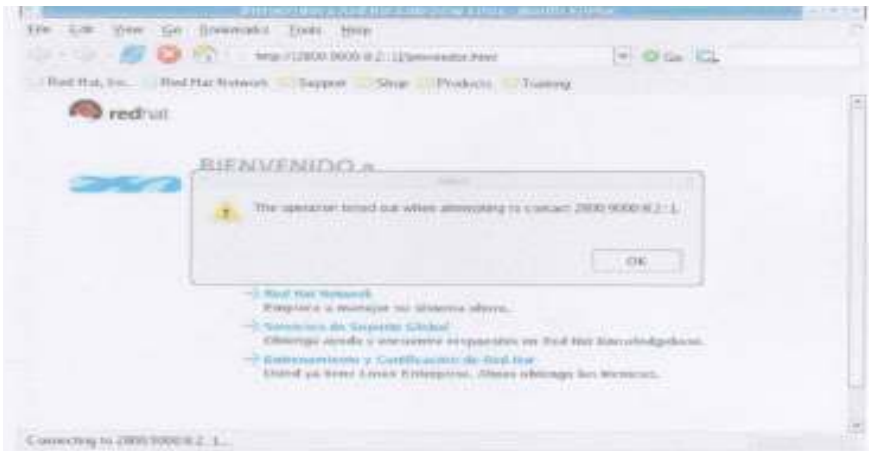


Figura 87. La Pc3 y PC4 no pueden ingresar al sistema de la otra Red ni al servicio FTP.

10.9 Guía de implementación de Calidad de Servicio (QoS) en Linux

El actual kernel de Linux, tiene características que permite realizar un control avanzado del tráfico aplicando QoS (Calidad de Servicio). Se puede encontrar dos protocolos que realizan dicha tarea: Diffserv (RFC 2475) y RSVP. Se entiende por QoS el poder tratar a un paquete u otro dependiendo de sus características, con lo que se obtiene una una mejor gestión del ancho de banda.

Para una buena gestión del ancho de banda, se necesita el paquete iproute2 e ip6tables.

El comando ip link show, muestra los "enlaces" presentes en una computadora, muestra la información de la interfaz ethernet y su correspondiente MTU (Maximum transmission unit) que es 1500 a esto se le llama disciplinas de colas (qdisc).

```
[root@root ~]# ip link show
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000
   link/ether 00:0c:76:ee:fe:a7 brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000
   link/ether 00:08:a1:a5:4c:c8 brd ff:ff:ff:ff:ff:ff
4: sit0: <NOARP> mtu 1480 qdisc noop
   link/sit 0.0.0.0 brd 0.0.0.0
```

Cuando se envían datos por una interfaz, por defecto se encolan en una FIFO (el primero que entra es el primero que sale), y cuando se tiene una gran cantidad de tráfico, si se suma el tiempo que tarda el paquete en entrar y salir de la FIFO y repetir el mismo proceso; se obtiene como resultado latencia.

Disciplinas de colas con clase

Poseen un nivel de profundidad. Es decir, se puede tener una **qdisc** raíz de la que cuelguen ciertas subdivisiones, en las cuales se puede agrupar diferentes tipos de paquetes (clases) para luego asignarles prioridades. A su vez, las clases pueden tener otras qdisc con o sin clases. Por defecto, cuando se crea una clase, se le adjunta una qdisc FIFO (se puede cambiar por otra), al menos que de esa clase tenga clases hijas. Ver figura siguiente.

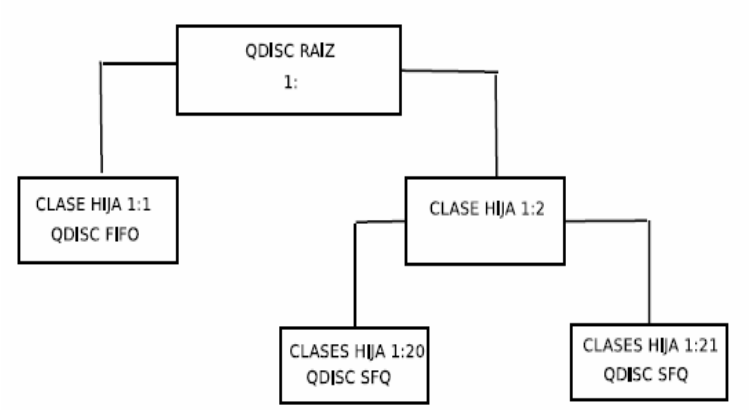


Figura 88. Estructura de una disciplinas de colas con clase.

Para identificar las clases, se utiliza una notación de dos números separados por dos puntos que indican mayor: menor. El mayor, se utiliza para referenciar la raíz de una clase, en cambio el menor se refiere a cualquiera de las clases que descienden de la raíz padre.

Para tratar a los paquetes en las diferentes clases, se debe de utilizar un filtro, de la misma forma que se construyen las reglas del firewall, en caso de cumplir unas determinadas características, irán a unas clases o a otras. Las principales qdisc con clases son las siguientes:

• **CBQ:** Algoritmo que basándose en prioridades, envía paquetes. Útil para enviar paquetes a clases hijas.

• **HTB:** es una cola multibanda porque permite generar varias colas "paralelas". Por ejemplo se puede crear 4 colas, de la 0 a la 3. Es jerárquica porque acepta clasificación de paquetes por clases.

10.9.1 Filtros para QoS

Los filtros son los que permiten clasificar los paquetes en diferentes clases:

• **fw**: ***clasifica el trafico, se basa en la marca del paquete que se le colocaron ip6tables utilizando el comando MARK.***

• **u32**: Se basa en las características de la cabecera IPv6.

Reservando el ancho de banda según protocolo para ello se asingaran prioridades a los protocolos:

FTP -> prio1

IPv6, http->prio2

```
[root@root ~]# tc qdisc add dev eth1 root handle 1: htb default 1
# Se agrega una disciplina de cola en la interfaz eth1, se establece el nodo principal con la instrucción root handle1: htb default 1. Htb indica que es una cola multibanda.

[root@root ~]# tc class add dev eth1 parent 1: classid 1:1 htb rate 120kbit ceil 120kbit
# Con esta instrucción se crea la clase padre en la interfaz eth1, su didentificador es 1:1 el ancho de banda total que posee es 120 kbit.

[root@root ~]# tc class add dev eth1 parent 1:1  classid 1:10 htb rate 100kbit ceil 120kbit prio 1
#  La instrucción anterior crea una clase hija 1:1 con identificación 1:10 y utiliza 100 kbit de los  120 kbit disponible, se le ha asignado la prioridad uno, es decir que esta división del ancho de banda se ocupa para el tráfico FTP.

[root@root ~]# tc class add dev eth1 parent 1:1 classid 1:11 htb rate 20kbit ceil 120kbit prio 2
#  La instrucción anterior crea una clase hija 1:1 con identificación 1:11 y utiliza 20 kbit de los  120 kbit disponible, se le ha asignado la prioridad dos, es decir que esta división del ancho de banda se ocupa para el tráfico IPv6 y para http.

[root@root ~]# tc qdisc add dev eth1 parent 1:10 handle 100: sfq perturb 10 quantum 1500
#  En este caso se reserva el ancho de banda con la regla sfq (stochactic Fairness Queueing) esta reserva pertenece a la clase hija  con identificación 1:10, la cual pertenece a la prioridad uno para el tráfico FTP.

[root@root ~]# tc qdisc add dev eth1 parent 1:11 handle 110: sfq perturb 10 quantum 1500
# Se reserva el ancho de banda con la regla sfq, pertenece a la clase hija  con identificación 1:11, la cual pertenece a la prioridad uno para el tráfico IPv6 y http.

[root@root ~]# tc filter add dev eth1 parent 1:0 protocol ipv6 prio 1 handle 1 fw classid 1:10
# Se agrega en la interfaz eth1 la clase 1:0 que es la padre, para que el protocolo ipv6 pueda manejar también la clase hija que tiene prioridad uno, y su identificación es 1:10, que es para el tráfico FTP
```

```
[root@root ~]# tc filter add dev eth1 parent 1:0 protocol ipv6 prio 2 handle 2 fw classid 1:11
# Se agrega en la interfaz eth1 la clase 1:0 que es la padre, para que el protocolo ipv6 pueda manejar también la clase hija que tiene prioridad uno, y su identificación es 1:11, que es para el tráfico IPv6 y http.
```

Ahora se marca los puertos

FTP tiene prioridad uno

```
[root@root ~]# ip6tables -t mangle -A PREROUTING -p tcp --dport 20:21 -j MARK --set-mark 1
# Se marcan los paquetes antes de que entren a la tabla de ruteo, el parámetro prerouting pertenece a la tabla mangle de ip6tables. El puerto que se está marcando es 20:21 que se utiliza para el servicio FTP, y se coloca set mark 1, por que éste tráfico tiene prioridad uno.

[root@root ~]# ip6tables -t mangle -A PREROUTING -p tcp --dport 20:21 -j RETURN

[root@root ~]# ip6tables -t mangle -A OUTPUT -p tcp --dport 20:21 -j MARK --set-mark 1
# Se marcan los paquetes salientes del firewall que utilizan el puerto 20:21 para FTP.

[root@root ~]# ip6tables -t mangle -A OUTPUT -p tcp --dport 20:21 -j RETURN
```

IPcmpv6 tiene prioridad dos

```
[root@root ~]# ip6tables -t mangle -A PREROUTING -p icmpv6 -j MARK --set-mark 2
# Se marcan los paquetes antes de que entren a la tabla de ruteo, el parámetro prerouting pertenece a la tabla mangle de ip6tables. El puerto que se está marcando es el protocolo icmpv6, y se coloca set mark 2, por que éste tráfico tiene prioridad dos.

[root@root ~]# ip6tables -t mangle -A PREROUTING -p icmpv6 -j RETURN

[root@root ~]# ip6tables -t mangle -A OUTPUT -p icmpv6 -j MARK --set-mark 2
# Se marcan los paquetes salientes del firewall que utilizan el protocolo icmpv6

[root@root ~]# ip6tables -t mangle -A OUTPUT -p icmpv6 -j RETURN
```

HTTP tiene prioridad dos

```
[root@root ~]# ip6tables -t mangle -A PREROUTING -p tcp --dport 80 -j MARK --set-mark 2
# Se marcan los paquetes antes de que entren a la tabla de ruteo, el parámetro prerouting pertenece a la tabla mangle de ip6tables. El puerto que se está marcando es 80 que se utiliza para el servicio HTP, y se coloca set mark 2, por que éste tráfico tiene prioridad dos.


[root@root ~]# ip6tables -t mangle -A PREROUTING -p tcp --dport 80 -j RETURN

[root@root ~]# ip6tables -t mangle -A OUTPUT -p tcp --dport 80 -j MARK --set-mark 2
# Se marcan los paquetes salientes del firewall que utilizan el puerto 80 para HTTP.

[root@root ~]# ip6tables -t mangle -A OUTPUT -p tcp --dport 80 -j RETURN

[root@root ~]# ip6tables -t mangle -A PREROUTING -p tcp --dport 443 -j MARK --set-mark 2
# Se marcan los paquetes antes de que entren a la tabla de ruteo, el parámetro prerouting pertenece a la tabla mangle de ip6tables. El puerto que se está marcando es 443 que se utiliza también para el servicio HTP, y se coloca set mark 2, por que éste tráfico tiene prioridad dos.
```

```
[root@root ~]# ip6tables -t mangle -A PREROUTING -p tcp --dport 443 -j RETURN

[root@root ~]# ip6tables -t mangle -A OUTPUT -p tcp --dport 443 -j MARK --set-mark 2
# Se marcan los paquetes salientes del firewall que utilizan el puerto 443 que se utiliza también para el servicio HTTP

[root@root ~]# ip6tables -t mangle -A OUTPUT -p tcp --dport 443 -j RETURN
```

El parámetro **RETURN** se utiliza para la verificación del paquete contra las reglas de la cadena actual. Si el paquete con un destino RETURN cumple una regla de una cadena llamada desde otra cadena, el paquete es devuelto a la primera cadena para retomar la verificación de la regla allí donde se dejó.

Se guardan las reglas

```
[root@root ~]# service ip6tables save
# Luego se guardan las cadens con service ip6tables save

Guardando las reglas del cortafuegos a :                [  OK  ]
```

Se pueden verificar las reglas

```
[root@root ~]# service ip6tables status
# La instrucción service ip6tables, se utiliza para ver el estado de las cadenas definidas, como se muestra a continuación aparecen las reglas definidas en la sección anterior, que solo se utilizan para restringir el tráfico entre las dos redes y adicionalmente también aparecen las que se agregaron para marcar el tráfico que pasa por la interfaz eth1.


Tabla: filter
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all      2800:9000:8:2::2/128  2800:9000:8:1::1/128
DROP       all      2800:9000:8:2::3/128  2800:9000:8:1::1/128
DROP       all      2800:9000:8:1::2/128  2800:9000:8:2::1/128
DROP       all      2800:9000:8:1::3/128  2800:9000:8:2::1/128
DROP       tcp      2800:9000:8:2::2/128  2800:9000:8:2::1/128tcp dpts:20:21

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
DROP       all      2800:9000:8:1::/64   2800:9000:8:2::/64

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Tabla: mangle
Chain PREROUTING (policy ACCEPT)
Target  prot      opt source        destination
MARK       tcp        ::/0              ::/0            tcp dpts:20:21 MARK set 0x1
RETURN   tcp         ::/0              ::/0             tcp dpts:20:21
```

```
MARK       icmpv6  ::/0                ::/0             MARK set 0x2
RETURN     icmpv6   ::/0               ::/0
MARK       tcp        ::/0             ::/0             tcp dpt:80 MARK set 0x2
RETURN     tcp        ::/0             ::/0             tcp dpt:80
MARK       tcp        ::/0             ::/0             tcp dpt:443 MARK set 0x2
RETURN     tcp        ::/0             ::/0             tcp dpt:443

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot       opt source        destination
MARK       tcp          ::/0              ::/0             tcp dpts:20:21 MARK set 0x1
RETURN     tcp          ::/0              ::/0             tcp dpts:20:21
MARK       icmpv6     ::/0              ::/0             MARK set 0x2
RETURN     icmpv6     ::/0                ::/0
MARK       tcp      ::/0              ::/0             tcp dpt:80 MARK set 0x2
RETURN     tcp      ::/0              ::/0             tcp dpt:80
MARK       tcp      ::/0              ::/0             tcp dpt:443 MARK set 0x2
RETURN     tcp      ::/0              ::/0             tcp dpt:443

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
```

Si se desea ver las clases asignadas en la eth1 se utiliza el siguiente comando:

```
[root@root ~]# tc -s class show dev eth1
class htb 1:11 parent 1:1 leaf 110: prio 2 rate 20Kbit ceil 20Kbit burst 1601b cburst 1601b
 Sent 0 bytes 0 pkts (dropped 0, overlimits 0 requeues 0)
 lended: 0 borrowed: 0 giants: 0
 tokens: 656179 ctokens: 656179

class htb 1:1 root rate 120Kbit ceil 120Kbit burst 1614b cburst 1614b
 Sent 0 bytes 0 pkts (dropped 0, overlimits 0 requeues 0)
 lended: 0 borrowed: 0 giants: 0
 tokens: 110249 ctokens: 110249

class htb 1:10 parent 1:1 leaf 100: prio 1 rate 100Kbit ceil 100Kbit burst 1611b cburst 1611b
 Sent 0 bytes 0 pkts (dropped 0, overlimits 0 requeues 0)
 lended: 0 borrowed: 0 giants: 0
 tokens: 132055 ctokens: 132055
```

Verificando las reglas de las colas:

```
[root@root ~]# tc qdisc show dev eth1
qdisc htb 1: r2q 10 default 1 direct_packets_stat 27
qdisc sfq 100: parent 1:10 limit 128p quantum 1500b perturb 10sec
qdisc sfq 110: parent 1:11 limit 128p quantum 1500b perturb 10sec
```

Verificando el filtro tc asignado en la interfaz eth1

```
[root@root ~]# tc filter show dev eth1
filter parent 1: protocol ipv6 pref 1 fw
filter parent 1: protocol ipv6 pref 1 fw handle 0x1 classid 1:10
filter parent 1: protocol ipv6 pref 2 fw
filter parent 1: protocol ipv6 pref 2 fw handle 0x2 classid 1:11
```

Verificando las reglas de colas por defecto en la interfaz eth1

```
[root@root ~]# tc -s qdisc ls dev eth0
qdisc pfifo_fast 0: bands 3 priomap  1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1
 Sent 3026 bytes 17 pkts (dropped 0, overlimits 0 requeues 0)
```

Para borrar las listas de colas se utiliza el siguiente comando:

```
root@root ~]#tc qdisc del dev eth0 root
```

10.10 Guía de configuración de una VPN.

Red Hat Enterprise Linux es compatible con IPsec para la conexión entre hosts y redes remotos utilizando un túnel seguro en un transportador de red común tal como la Internet. IPsec se puede implementar usando una conexión host-a-host (una computadora a la otra) o de red-a-red (una LAN/WAN a la otra). La implementación IPsec en Red Hat se realiza a través de los demonios Raacon o Pluto, los cuales se encuentran en parte experimental. Para emplear una conexión punto a punto y simular una VPN se utiliza mecanismos de transición.

10.10.1 Mecanismos de Transición

IPv6 ha sido diseñado de tal forma que se facilite la transición y coexistencia con Ipv4. Se han diseñado diferentes estrategias para la coexistencia con redes/nodos Ipv4. Los Mecanismos de transición son los siguientes:

- Doble pila, o soporte simultáneo de IPv4 e IPv6.
- Traducción IPv4/IPv6 como último recurso, dado que no es perfecto.
- Túneles, o encapsulado de IPv6 sobre IPv4 (y viceversa).

Doble Pila

Las características principales de éste mecanismo son:

- Los nodos tienen implementadas las pilas Ipv4 e Ipv6.
- Comunicaciones con nodos solo Ipv6, que implica pila ipv6 asumiendo soporte Ipv6 en la red.
- Comunicaciones con nodos solo ipv4, involucra solamente a la pila IPv4.

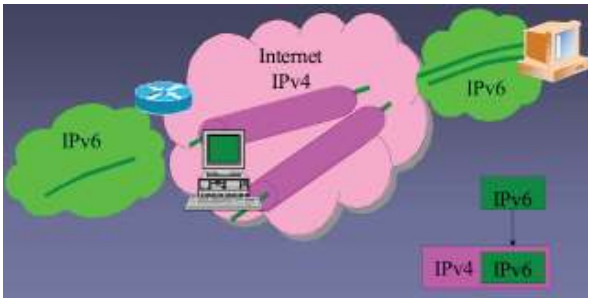


Figura 89. Esquema de doble pila.

Túneles Ipv6 sobre Ipv4

Existen diversos mecanismos de transición basados en túneles, cada uno con una forma diferente de encapsulacion

Figura 90. Tipos de encapsulación

10.10.2 Túneles IPv6 sobre IPv4

Algunos mecanismos de transición basados en túneles

- **6in4**
- TB
- TSP
- 6to4
- Teredo
- Túneles automáticos
- ISATAP
- 6over4

10.10.2.1 Túneles 6in4

Características

- Encapsula directamente el paquete ipv6 dentro de un paquete ipv4.
- Se suele hacer entre: dos routers, o dos nodos finales.
- El túnel se considera como un enlace punto a punto desde el punto de vista de IPv6
- Solo un salto Ipv6 aunque existan varios Ipv4.
- Las direcciones IPv6 de ambos extremos del túnel son del mismo prefijo
- Todas las direcciones IPV6 del nodo final siempre pasan por el router que está en el extremo final del túnel.

- Los túneles 6in4 pueden construirse desde nodo finales situados detrás de NAT

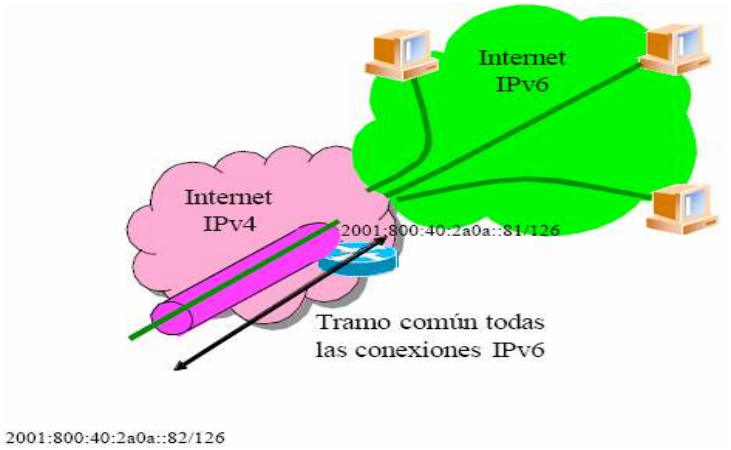


Figura 91. Esquema 6 in 4

10.10.2.2 Túnel Broker

Para facilitar la asignación de direcciones y creación de túneles ipv6, se ha desarrollado el concepto de Túnel Broker (TB).

Que es un intermediario al que el usuario final se conecta, normalmente con una interfaz Web. El usuario solicita al TB la creación de un túnel y este le asigna una dirección IPv6 y le proporciona instrucciones para crear el túnel en el lado del usuario.

El Túnel Broker también configura el router que presenta el extremo final del túnel para el usuario.

10.10.2.3 Túneles 6to4

Se trata de un encapsulado de paquetes IPv6 en paquetes Ipv4, similar a 6in4

Diferencias:

- La dirección ipv6 del cliente no depende del router al que se conecta, sino de la dirección ipv4 pública: Rango 2002::/16
- Los paquetes IPv6 de salida del cliente siempre son enviados al mismo “6to4 relay”, sin embargo, los paquetes IPv6 de entrada al cliente pueden provenir de otros “6to4 relay” diferentes.

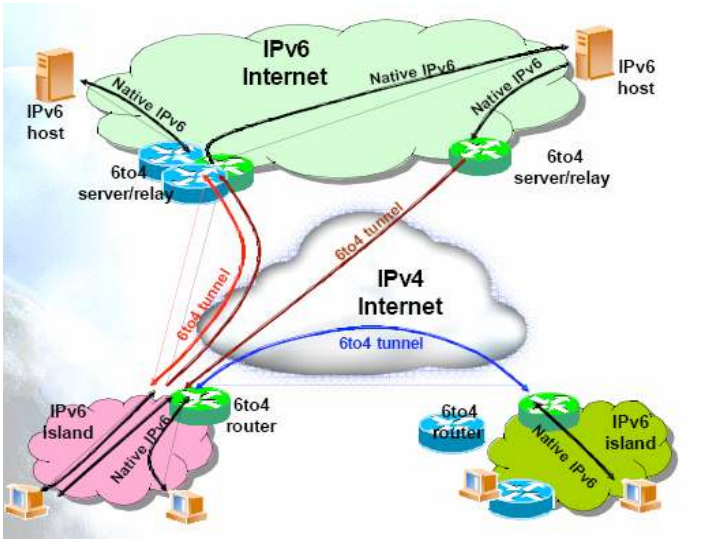


Figura 92. Esquema 6 to 4.

10.10.2.4 Túneles Teredo

Teredo está diseñado para proporcionar IPv6 a nodos que están ubicados detrás de NAT

Existen diferentes tipos: teredo Server, teredo relay, teredo cliente.

El cliente configura un Teredo Server que le proporciona una dirección ipv6 del rango **3FFE:831F::/32** basada en la dirección ipv4 pública y el puerto usado.

Si el Teredo Server configurado es además Teredo Relay, el cliente tiene conectividad IPv6 con cualquier nodo IPv6. De lo contrario solo se tiene conectividad IPv6 con otros clientes teredo,

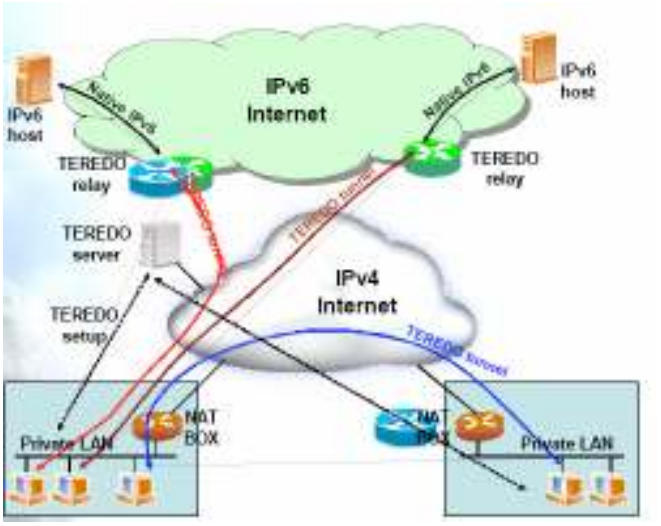


Figura 93. Esquema Túnel Teredo.

10.10.3 Establecimiento túnel 6in4

Scripts de creación de túneles 6in4

a) Windows XP/2003 (desde ventana de comandos)

netsh interface ipv6 add v6v4tunnel "Tunel01" direccion_ipv4_local direccion_ipv4_remota

netsh interface ipv6 add address "Tunel01" direccion_ipv6

netsh interface ipv6 add route ::/0 "Tunel01" direccion_gateway_ipv6 publish=yes

b) Linux (desde la ventana de commandos)

ip tunnel add Tunel01 mode remote direccion_ipv4_remote local direccion_ipv4_local ttl 255

ip link set Tunel01 up

ip addr add direccion_ipv6/126 dev Tunel01

ip route add 2000::/3 dev Tunel01

Cuando en el router (PC principal), se activa el servicio de ip6tables, entonces se filtran los paquetes, y se activas las reglas previamente configuradas, en este caso, los datos no pueden pasar de una subred a otra

Los túneles permiten la comunicación entre dos nodos, simulando como que si estuvieran en la misma red, y se puede acceder fácilmente, no importando las reglas del firewall. El túnel permite la función principal de una VPN, que es conectarse remotamente y directamente a cierta Red

.

Para este caso, la configuración se aplicará para el nodo PC3 y Pc1, ambas pertenecen a redes distintas. Ver figura siguiente:

|

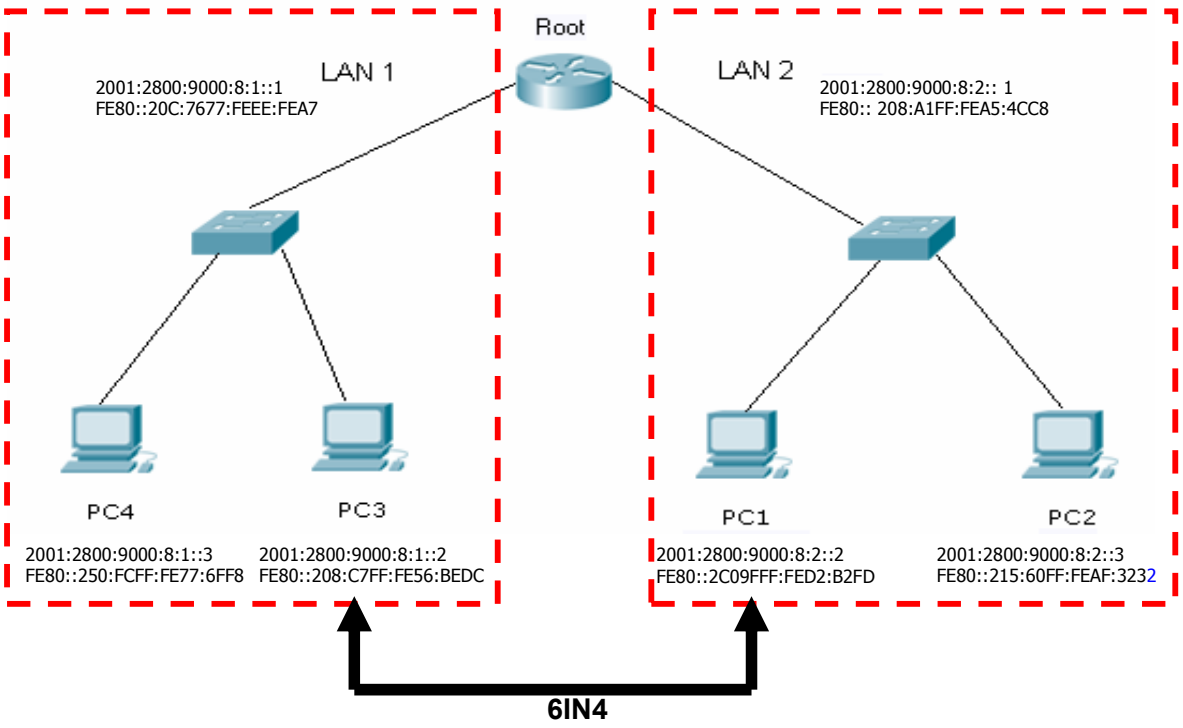


Figura 94. En este esquema Lan1 no se puede comunicar con la Lan2, solamente la pc1 y pc3 a través de un túnel 6in4.

Datos

PC3 (Red Hat)

Dirección ipv4 local	192.168.20.3
Dirección ipv4 remota	192.168.30.3
Dirección ipv6	2001:10:20:30::12/126
Dirección de puerta de enlace ipv6	2001:10:20:30:11/126

PC1 (Knoppix)

Dirección ipv4 local	192.168.30.3
Dirección ipv4 remota	192.168.20.3
Dirección ipv6	2001:10:20:30::11/126
Dirección de puerta de enlace ipv6	2001:10:20:30:12/126

10.10.4 Configuración de 6in4

PC3 (Red Hat)

```
[root@root ~]# ip tunnel add Tunel01 mode sit remote 192.168.30.3 local 192.168.20.3 ttl 255
#  Con la instrucción ip tunnel, se crea el tunel llamado Tunel01, se coloca la dirección remota (192.168.30.3) y se especifica la dirección local que es 192.168.20.3

[root@root ~]# ip link set Tunel01 up
# Tunel01 es ahora una interfaz, y con ip link set Tunel01 up, se levanta para que pueda ser configurado correctamente.

[root@root ~]# ip addr add 2001:10:20:30::12/126 dev Tunel01
# Ahora se procede a asignarle una dirección ipv6, en este caso es distinta a la configurada en la interfaz eth1, que es la 2800:9000:8:1::2.

[root@root ~]# ip route add 2000::/3 dev Tunel01
# Luego se agrega la ruta por defecto.
```

Para verificar la configuración se utiliza **Ifconfig**

```
[root@root ~]# ifconfig Tunel01
Tunel01   Link encap:IPv6-in-IPv4
          inet6 addr: 2001:10:20:30::12/126 Scope:Global
          inet6 addr: fe80::c0a8:1403/128 Scope:Link
          UP POINTOPOINT RUNNING NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

PC1 (Knoppix)

```
root@Knoppix: ~# ip tunnel add Tunel01 mode sit remote 192.168.20.3 local 192.168.30.3 ttl 255
#  Con la instrucción ip tunnel, se crea el tunel llamado Tunel01, se coloca la dirección remota (192.168.20.3) y se especifica la dirección local que es 192.168.30.3

root@Knoppix: ~# ip link set Tunel01 up
```

```
# Tunel01 es ahora una interfaz, y con ip link set Tunel01 up, se levanta para que pueda ser configurado correctamente.

root@Knoppix: ~#  ip addr add 2001:10:20:30::11/126 dev Tunel01
# Ahora se procede a asignarle una dirección ipv6, en este caso es distinta a la configurada en la interfaz eth1, que es la 2800:9000:8:2::2.

root@Knoppix: ~#  ip route add 2000::/3 dev Tunel01
# Luego se agrega la ruta por defecto.
```

Para verificar la configuración se utiliza **Ifconfig**

```
root@Knoppix: ~#  ifconfig Tunel01
Tunel01   Link encap:IPv6-in-IPv4
          inet6 addr: 2001:10:20:30::11/126 Scope:Global
          inet6 addr: fe80::c0a8:1e03/128 Scope:Link
          UP POINTOPOINT RUNNING NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Una vez configurado el túnel, PC1 y PC3 se pueden comunicar sin ningún problema:
En las figuras siguiente se muestra como la PC1 puede hacer ping6, a la dirección global, dirección de enlace y como puede acceder al servicio http configurado en la Pc2, a pesar que ambas computadoras no pertenecen a la misma red, y las reglas de ip6tables no permitan su comunicación.

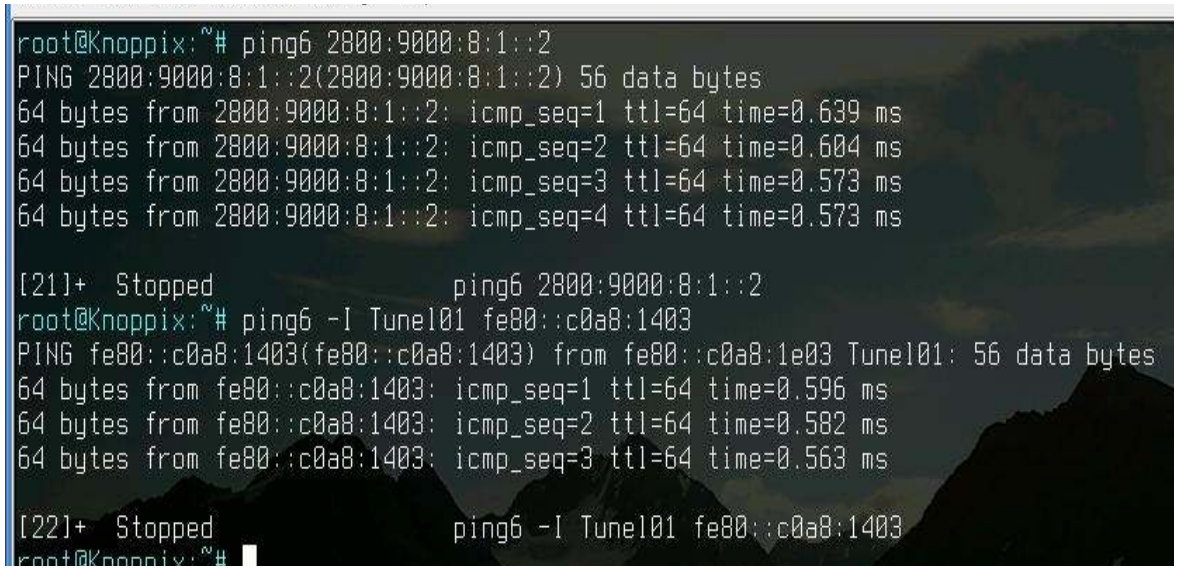


Figura 95. Haciendo ping6 a la dirección global y local de la PC3 desde la PC1

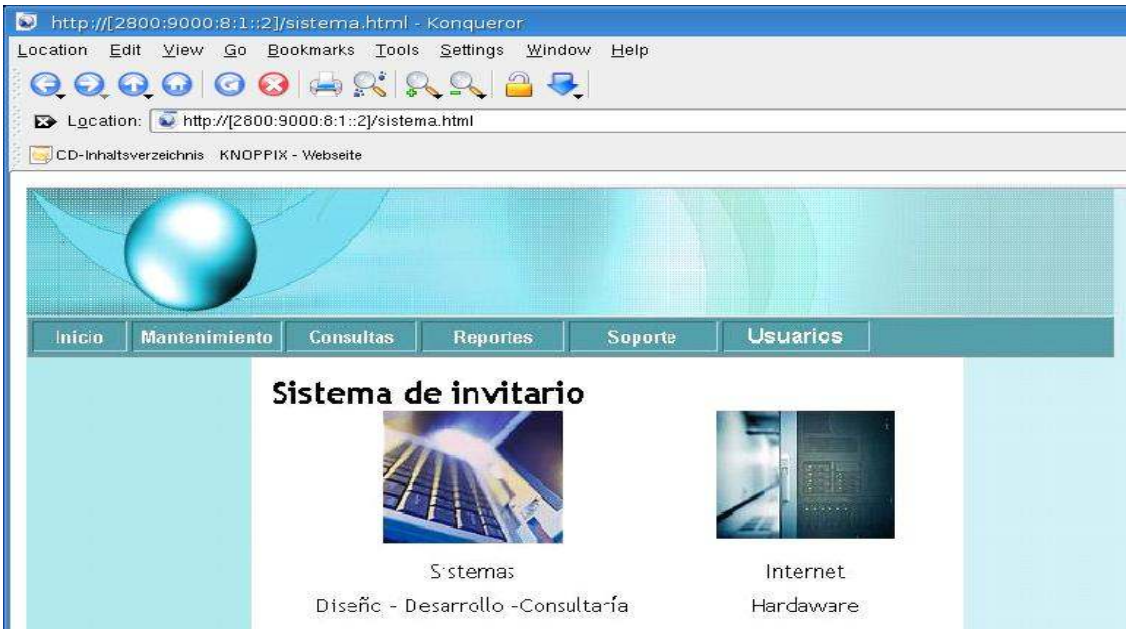


Figura 96. Desde la Pc1 se puede ingresar al servicio http de la Pc3

Capítulo

11

DNS IPv6

Temas:

- Configuración de DNS para IPv6
- Componentes de un DNS
- Zonas
- Ejemplo de configuración

11. Configuración de DNS para Ipv6 en Linux

El método que se tiene, en general, para referirse a un host, es a través del uso de literales o “nombres” y con ello se hace una mención implícita a la dirección IP del mismo. Una de las formas, de llevar a cabo este método “automático” para el usuario final, es mediante la implementación de un servicio conocido como “**DNS**” (Domain Name System). En primera instancia, el DNS fue definido para IPv4 mediante las especificaciones descritas en los RFC1034 y RFC1035. Dichos documentos fueron actualizados con el RFC1886 para ampliar el servicio a la resolución de direcciones con formato IPv6.

El RFC1886 define un tipo de registro denominado AAAA, que logra cumplir con el objetivo de llevar a cabo búsquedas basadas en IPv6. No obstante, focalizando las metas en hacer más fácil el método de Renumeración y Multi-proveedor, se ha desarrollado un documento (RFC2874) que propone la introducción de un nuevo tipo de registro: A6.

DNS es una base de datos distribuida y jerárquica que almacena la información necesaria para los nombres de dominio. Sus usos principales son la asignación de nombres de dominio a direcciones IP. El DNS nació de la necesidad de facilitar a los seres humanos el acceso hacia los servidores disponibles a través de Internet permitiendo hacerlo por un nombre, algo más fácil de recordar que una dirección IP**.**

En Linux se utiliza el demonio BIND (Berkeley Internet Name Domain) para configurar DNS. BIND es una implementación del protocolo DNS y provee una implementación libre de los principales componentes del Sistema de Nombres de Dominio, los cuales incluyen:

- Un servidor de sistema de nombres de dominio (named).
- Una biblioteca resolutoria de sistema de nombres de dominio.
- Herramientas para verificar la operación adecuada del servidor DNS (bind-utils).

Los Servidores DNS utilizan TCP y UDP en el puerto 53 para responder las consultas. Casi todas las consultas consisten de una sola solicitud UDP desde un Cliente DNS seguida por una sola respuesta UDP del servidor.

11.1 Componentes de un DNS.

Los DNS operan a través de tres componentes: Clientes DNS, Servidores DNS y Zonas de Autoridad.

a) Clientes DNS.

Son programas que ejecuta un usuario y que generan peticiones de consulta para resolver nombres. Básicamente preguntan por la dirección IP que corresponde a un nombre determinado.

b) Servidores DNS.

Son servicios que contestan las consultas realizadas por los **Clientes DNS**. Hay dos tipos de servidores de nombres:

1. Servidor Maestro (o primario)
Obtiene los datos del dominio a partir de un archivo hospedado en el mismo servidor.
2. Servidor Esclavo (o secundario)
Al iniciar obtiene los datos del dominio a través de un Servidor Maestro (o primario), realizando un proceso denominado transferencia de zona.

Los Servidores DNS responden dos tipos de consultas:

• **Consultas Iterativas (no recursivas)**
El cliente hace una consulta al Servidor DNS y este le responde con la mejor respuesta que pueda darse basada sobre su caché o en las zonas locales. Si no es posible dar una respuesta, la consulta se reenvía hacia otro Servidor DNS repitiéndose este proceso hasta encontrar al Servidor DNS que tiene la Zona de Autoridad capaz de resolver la consulta.

• Consultas Recursivas

El Servidor DNS asume toda la carga de proporcionar una respuesta completa para la consulta realizada por el Cliente DNS. El Servidor DNS desarrolla entonces Consultas Iterativas separadas hacia otros Servidores DNS (en lugar de hacerlo el Cliente DNS) para obtener la respuesta solicitada.

c) Zonas de Autoridad.

Permiten al Servidor Maestro o Primario cargar la información de una zona. Cada Zona de Autoridad abarca al menos un dominio y posiblemente sus sub-dominios, si estos últimos no son delegados a otras zonas de autoridad. La información de cada Zona de Autoridad es almacenada de forma local en un archivo en el Servidor DNS. Este archivo puede incluir varios tipos de registros:

Tipo de Registro.	Descripción.
A (Address)	Registro de dirección que resuelve un nombre de anfitrión hacia una dirección IPv4 de 32 bits.
A6 ó AAAA	Registro de dirección que resuelve un nombre de anfitrión hacia una dirección IPv6 de 128 bits.
CNAME (Canonical Name)	Registro de nombre canónico que hace que un nombre sea alias de otro. Los dominios con alias obtienen los sub-dominios y registros DNS del dominio original.
MX (Mail Exchanger)	Registro de servidor de correo que sirve para definir una lista de servidores de correo para un dominio, así como la prioridad entre éstos.
PTR (Pointer)	Registro de apuntador que resuelve direcciones IPv4 hacia el nombre anfitriones. Es decir, hace lo contrario al registro A. Se utiliza en zonas de Resolución Inversa.
NS (Name Server)	Registro de servidor de nombres que sirve para definir una lista de servidores de nombres con autoridad para un dominio.

SOA (Start of Authority)	Registro de inicio de autoridad que especifica el Servidor DNS Maestro (o Primario) que proporcionará la información con autoridad acerca de un dominio de Internet, dirección de correo electrónico del administrador, número de serie del dominio y parámetros de tiempo para la zona.
SRV (Service)	Registro de servicios que especifica información acerca de servicios disponibles a través del dominio. Protocolos como SIP (Session Initiation Protocol) y XMPP (Extensible Messaging and Presence Protocol) suelen requerir registros SRV en la zona para proporcionar información a los clientes.
TXT (Text)	Registro de texto que permite al administrador insertar texto arbitrariamente en un registro DNS.

Tabla 14 Componentes del archivo de una Zona de autoridad.

11.2 Zonas que se pueden resolver

a) Zonas de Reenvío.

Devuelven direcciones IP para las búsquedas hechas para nombres FQDN (Fully Qualified Domain Name). En el caso de dominios públicos, la responsabilidad de que exista una Zona de Autoridad para cada Zona de Reenvío corresponde a la autoridad misma del dominio, es decir, y por lo general, quien esté registrado como autoridad del dominio tras consultar una base de datos WHOIS.

b) Zonas de Resolución Inversa.

Devuelven nombres FQDN (Fully Qualified Domain Name) para las búsquedas hechas para direcciones IP. En el caso de segmentos de red públicos, la responsabilidad de que exista de que exista una Zona de Autoridad para cada Zona de Resolución Inversa corresponde a la autoridad misma del segmento, es decir, y por lo general, quien esté registrado como autoridad del segmento tras consultar una base de datos WHOIS.

11.3 Características de DNS para IPv6

Para IPv6, los nuevos archivos y zonas principales para reversas son definidos de la siguiente forma:

- **AAAA y reversa IP6.INT**: especificado en la RFC 1886 / DNS, Extensiones soportadas para IP versión 6, utilizada desde la versión 8 de la herramienta BIND.
- **A6, DNAME y reversa IP6.ARPA**: especificado en la RFC 2874/ DNS. Se utiliza a partir de la versión 9 del demonio BIND.

Definición en un archivo de zonas

Se denomina "archivo de zonas" del servidor de nombres a un **archivo de base de datos** que contiene los registros (cada una de las líneas del archivo) para resolver las distintas partes del dominio de la que es responsable. Es decir, es un archivo que contiene los datos para poder resolver las peticiones de nombres, asociadas a determinado dominio, en direcciones IP.

Uno de los registros que se define en un archivo de zona es el denominado "Registro de Dirección" (Registros de tipo A, ver RFC1034). Un Registro de Dirección sirve para asociar nombres de host a direcciones IP dentro de una zona o dominio. Éstos son los registros que componen la mayor parte del archivo de zona.

Estos registros tendrán un formato que será diferente ya sea que se trate de una declaración para resolver direcciones IPv6 o para resolver direcciones IPv4. Si se trata de IPv4, su formato es el siguiente:

<nombrehost> IN A <direcciónIPdehost>

Ejemplos:
machine1 IN A 157.55.201.143
Nombreservidor2 IN A 157.55.200.2

Lo que significa, por ejemplo, que un cliente no va a necesitar acordarse de digitar la dirección 157.55.201.143, sino que con tan solo escribir “machine1”, el servidor de DNS hará la traducción correspondiente basándose en el archivo de zona y en los Registros de Direcciones que allí se alojan.

En el caso de direcciones IPv6, existen los denominados Registros de Dirección AAAA o A6, que se utiliza de la siguiente forma:

<nombrehost> IN AAAA <direcciónIPdehost>

Ejemplos:
machine2 IN AAAA 3FFE:8070:1019:E00:1234::33
nombreservidor3 IN AAAA 3FFE:8070:1019:E00:1234::4

Existe un inconveniente cuando se cambia de proveedor y de prefijos, por consiguiente, se vuelve tediosa la tarea de actualizar todos y cada uno de los registros. Entonces, una manera más eficaz de hacer las traducciones, que no haga necesario el cambio en todas las zonas del dominio, sino que sea un sistema dinámico de traducción. Para ello se utilizan los registros de Dirección A6.

El objetivo de los Registros A6, no fue solamente facilitar la escritura, sino también agilizar los procesos de Renumbering y Multi-proveedor. Este tipo de registro tiene la particularidad de permitir que una consulta se haga en forma “recursiva”, es decir, que la respuesta a una petición no la proporcione un solo servidor, sino que la consulta puede “dividirse” en subconsultas y así, recursivamente, ir solicitando las distintas respuestas a los servidores correspondientes.

Para facilitar su comprensión, mostraremos el formato de la declaración de un Registro A6:
a.b.c A6 <NN> <address-suffix> <name>

Donde:
a.b.c es el nombre del dominio que se quiere resolver.
NN es el largo del prefijo, o sea, 128 - <address-suffix>
Address-suffix es la parte de la dirección que resuelve este registro.
Name es el próximo registro que resuelve la otra parte de la dirección

11.4 Ejemplo de configuración

1. Las configuraciones se realizan en el archivo "named.conf".

2. Se debe de tener por lo menos una zona en la cual se ubica los registros AAAA o A6 dependiendo de la versión de BIND que se instale. Si se tiene 8.x se trabaja con registros AAAA. Si se posee 9.x se utiliza registros A6.

3. En este ejemplo se utiliza la zona "ipv6.ejemplo.sv" y el archivo para esa zona es "named.ipv6". El dispositivo al cual se le configura una IPv6 es una computadora llamada "prueba".

4. En el archivo "named.ipv6" se configurara una de las siguientes líneas:
prueba.ipv6.ejemplo.sv. IN AAAA 2800:9000:8:1::2
Si es versión 9:
prueba.ipv6.ejemplo.sv. IN A6 0 2800:9000:8:1::2

5. Si se tiene la opción de contar con dominio inverso, se configura la zona para tener la resolución inversa e introducir los registros PTR[83]. En el caso de tener versiones 8.2.x soporta las zonas X.ip6.int y se tiene la posibilidad de usar registros DNAME. En caso de tener versiones 9.x soporta las zonas ip6.arpa.

Por ejemplo, si se tiene la zona para la delegación inversa de 3ffe:8070::/28. En el archivo "named.conf" se configura una de las siguientes zonas:

[83] *Registro de apuntador que resuelve direcciones IPv4 o IPv6*

```
zone "7.0.8.e.f.f.3.ip6.int" {
type master;
file "named.3ffe.807";
};
zone "\[x3FFE8070/28].ip6.arpa" {
type master;
file "named.3ffex807";
};
```

7. En el archivo de la zona "named.3ffe.807" se tiene:
1.9.0.0.b.a.e.f.f.f.8.0.0.6.2.0.0.0.0.0.0.2.0.0.0.0 IN PTR prueba.ipv6.ejemplo.sv.

En el archivo de la zona "named.3ffex807" se tiene:
\[x000020000026008fffeab0091/100] IN PTR prueba.ipv6.ejemplo.sv.

8 Es importante considerar que actualmente las versiones de BIND 8.2.x y 9.x, tanto los registros A6 como la zona IP6.ARPA, no son compatibles entre ellas. Por lo que si usa las versiones nuevas se tiene que tomar en cuenta las configuraciones soportadas para versiones 8.2.x. Si trabajas con versiones 8.2.x no podrás soportar los nuevos registros ni la nueva zona.

Capítulo

12

Protocolos de enrutamiento para IPv6

Temas:

- RIP
- OSPF
- BGP
- RIPng
- OSPFv3
- BGP-4

12. Protocolos de enrutamiento para ipv6.

Un protocolo de enrutamiento es el esquema de comunicación entre routers. Un protocolo de enrutamiento permite que un router comparta información con otros routers, acerca de las redes que conoce así como de su proximidad a otros routers. La información que un router obtiene de otro, mediante el protocolo de enrutamiento, es usada para crear y mantener las tablas de enrutamiento.

Protocolos de enrutamiento:

- Protocolo de información de enrutamiento (RIP)
- Protocolo "Primero la ruta más corta" (OSPF)
- Protocolo BGP (Border Gateway Protocol)

En las siguientes secciones se explicará como se utilizan estos protocolos con dispositivos CISCO y en el sistema operativo Linux a través del demonio Quagga.

12.1 Protocolos de enrutamiento en dispositivos CISCO

12.1.1 RIP Para Ipv6.

Los requerimientos del IOS se describen en la siguiente tabla.

Característica	Mínimo Cisco IOS Requerido
RIP para Ipv6	12.2(2)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T
Predistribución de Ruta	12.2(2)T, 12.0(21)ST, 12.0(22)S, 12.2(14)S, 12.3, 12.3(2)T

Tabla 15 Requerimientos de IOS para RIP Ipv6.

Las funciones que ofrece RIPv6 poseen los mismos beneficios que ofrece para Ipv4. Cada proceso de RIP en Ipv6 posee una tabla de enrutamiento local, que se encuentra referida a una base de datos de información de enrutamiento (RIB). La RIB contiene una tabla que posee los mejores costos de las rutas Ipv6 aprendidas por RIP. Si RIP Ipv6 aprende la misma ruta de dos vecinos diferentes, pero con costos diferentes, este almacenara en su tabla solamente la de menor costo en la RIB local.

Habilitando RIP para IPV6.

Para habilitar RIP Ipv6 debemos crear primeramente un proceso de enrutamiento RIP Ipv6 y especificarlo en la interface. Antes de habilitar RIP Ipv6, se debe habilitar el protocolo Ipv6 globalmente en el router utilizando el comando **ipv6 unicast-routing** y habilitar Ipv6 en cualquier interfase a la cual se le habilitara RIP.

A continuación se describen los pasos necesarios para habilitar RIP en una interfase.

PASO	COMANDO	PROPOSITO
1	**enable** **Ejemplo:** Router> enable	Habilita el modo EXEC privilegiado.
2	**configure terminal** **Ejemplo:** Router# configure terminal	Accesa al modo de configuración global.
3	**interface** interface-type interface-number **Ejemplo:** Router(config)# interface Ethernet 0/0	Especifica el número, el tipo de interface y entra al modo de configuración de interface.
4	**ipv6 rip** name **enable** **Ejemplo:** Router(config-if)# ipv6 rip process1 enable	Habilita el proceso de enrutamiento RIP Ipv6 sobre una interfase.

Tabla 16 Habilitando RIP en una Interfase.

12.1.2 OSPF para IPv6.

Prerrequisitos para Implementar OSPF para Ipv6.

Antes de implementar el protocolo de enrutamiento OSPF (Primero la Ruta Libre mas Corta) RFC2740, se debe hacer los siguiente:

- Planear la distribución y aplicación de OSPF sobre nuestra red. Por ejemplo, se debe decidir cual será el número de áreas requerido.
- Habilitar el enrutamiento unicast Ipv6.
- Habilitar Ipv6 sobre la interfase.
- Habilitar CEFv4 globalmente en el router utilizando el comando **ip cef** en el modo de configuración global.
- Habilitar CEFv6 globalmente en el router utilizando el comando **ipv6 cef** en el modo de configuración global.

La siguiente tabla muestra las últimas versiones de Sistema Operativo Cisco IOS que soportan las características de OSPF.

CARACTERISTICA	CISCO IOS REQUERIDO
Expansión de OSPF versión 3 sobre OSPF versión 2.	12.2(15)T, 12.0(24)S, 12.2(18)S, 12.3, 12.3(2)T
Tipos de LSA en OSPF para Ipv6.	12.2(15)T, 12.0(24)S, 12.2(18)S, 12.3, 12.3(2)T
Interfases NBMA en OSPF para Ipv6	12.2(15)T, 12.0(24)S, 12.2(18)S, 12.3, 12.3(2)T
Force SPF en OSPF para Ipv6	12.2(15)T, 12.0(24)S, 12.2(18)S, 12.3, 12.3(2)T
Balance de Carga en OSP para Ipv6	12.2(15)T, 12.0(24)S, 12.2(18)S, 12.3, 12.3(2)T
Direcciones sobre una interfase en OSPF para Ipv6	12.2(15)T, 12.0(24)S, 12.2(18)S, 12.3, 12.3(2)T
OSPF para soporte de autenticación Ipv6 con IPsec	12.3(4)T

Tabla 17 Cisco IOS para OSPF

Funcionamiento de OSPF.

OSPF es un protocolo de enrutamiento IP. Este es un protocolo de estado de enlace y no de vector distancia. Tomando en cuenta un enlace que pertenece a una interfase de un dispositivo de red; un protocolo de estado de enlace toma sus decisiones de enrutamiento de acuerdo al estado de los enlaces que conectan a un dispositivo origen y un destino. El estado de un enlace es la descripción de la interfase y su relación con los dispositivos de red vecinos.

La información de la interfase incluye el prefijo Ipv6 de la interfase, la mascara de red, el tipo de red a la que esta conectada, los routers conectados a la red, etc. Esta información es propagada a través de varios tipos de anuncios de estado de enlace (LSAs).
Una colección de datos LSAs de un router es almacenada en una base de datos de estado de enlace. El contenido de la base de datos. La diferencia entre la base de datos y la tabla de enrutamiento, es que la base de datos una colección completa de datos en bruto; la tabla de enrutamiento contiene una lista de las rutas más cortas hacia los destinos, conocidas a través de puertos específicos de la interfase de un router.

Tipos de LSAs.

La siguiente lista describe los diferentes tipos de LSA, donde cada una de ellas tiene un propósito diferente.

- **LSAs de Enrutamiento (Tipo 1)**

Describen el estado de enlace y los costos de los enlaces de un router para un área específica. Estas LSAs son inundadas dentro de una sola área. Estas también indican si un router es un Router de Borde de Área (ABR) o un Router de Limite de Sistema Autónomo (ASBR) y si este es un dispositivo final o un enlace virtual. En OSPF para Ipv6, estas LSAs no poseen información de dirección y son independientes del protocolo de red. La información de interfase de un router es expandida a través de múltiples LSAs y los receptores deben concatenar todas las LSAs recibidas cuando esta corriendo el algoritmo SPF.

- **LSAs de Red (Tipo 2)**

Describen el estado de enlace y la información de costo para todos los routers que forman parte de la red. Solamente un router designado puede generar una LSA de Red; las cuales no poseen información de dirección y son independientes del protocolo de red.

- **LSAs de Prefijos de Área Interna para ABRs (Tipo3)**

Anuncian redes internas a routers en otras áreas y representan a una sola red o a una conjunto de redes sumarizadas dentro de un solo anuncio. En OSPF para Ipv6 las direcciones para estas LSAs son representadas por un prefijo y la longitud del prefijo, en lugar de una dirección y su mascara de subred. Una ruta por defecto es representada por una longitud de prefijo igual a cero.

- **LSAs de Routers de Área Interna para ASBRs (Tipo 4)**

Anuncia la ubicación de un ASBR. Los routers que están tratando de alcanzar una red externa utilizan estos anuncios para determinar la mejor ruta para el siguiente salto.

- **LSAs de Sistema Autónomo Externo (Tipo 5)**

Redistribuyen rutas a otros sistemas autónomos, usualmente de un protocolo de enrutamiento diferente hacia OSPF. En OSPF para Ipv6 las direcciones para estas LSAs son representadas por un prefijo y la longitud del prefijo, en lugar de una dirección y su mascara de subred. Una ruta por defecto es representada por una longitud de prefijo igual a cero.

- **LSAs de Enlace (Tipo 8)**

Provee la dirección de enlace local de un router a los otros routers agregados al enlace y les informa sobre una lista de prefijos Ipv6 para ser asociados con el enlace y permite que el router establezca una colección de bits de opción para ser asociados con la LSA de red que será originada por el enlace.

- **LSAs para Prefijos de Área Interna (Tipo 9)**

Un router puede originar muchas de estas LSAs para cada router o red en transito, cada una con un identificador de estado de enlace único. Este identificador define para cada LSA la asociación que tenga con una LSA de Enrutamiento o LSA de Red y contiene los prefijos para la red stub o redes en transito.

NBMA en OSPF para Ipv6.

En una red NBMA, el router designado (DR) o el router designado de respaldo (BDR) son encargados de la inundación LSA. Sobre una red punto a punto, la inundación sale solamente de una interfase hacia un solo vecino. Los routers que comparten un segmento común (enlace de capa 2 entre dos interfases) comparten los vecinos sobre este segmento. OSPF utiliza el protocolo Hola (Hello), enviando periódicamente este tipo de paquetes a cada interfase.

En las redes punto a punto y redes punto a multipunto, el software inunda con actualizaciones de enrutamiento a los routers inmediatos. Solamente en los segmentos broadcast o NBMA, OSPF minimiza la cantidad de información que se intercambia en un segmento por medio de la elección de un Router Designado (DR) y un Router Designado de Respaldo (BDR), de esta manera los routers sobre el segmento tendrán un punto central de contacto para el intercambio de información. En lugar de que los routers intercambien actualizaciones de enrutamiento con cada uno de los routers del segmento, los routers del segmento intercambiaran información con el DR o el BDR, los cuales distribuirán esta información a los otros routers.

OSPF determina las prioridades de los routers sobre el segmento, para determinar cuales routers serán el DR y el BDR. El router con la prioridad más alta es seleccionado como DR. Un router con una prioridad igual a cero no puede ser elegido como DR o BDR.
Cuando se utiliza NBMA en OSPF para Ipv6, no es posible detectar vecinos automáticamente, por lo tanto los vecinos deben ser configurados manualmente en el modo de configuración del router.

Algoritmo SPF en OSPF para Ipv6.

Cuando la palabra **process** se utiliza con el comando **clear Ipv6 ospf,** la base de datos OSPF se vacía y se vuelve a calcular, entonces comienza a realizarse el algoritmo de Primero la Ruta Mas Corta (SPF). Cuando la palabra **force-spf** se utiliza con el comando **clear ipv6 ospf**, la base de datos no se vacía antes de que se realice el algoritmo SPF.

Balance de Carga en OSPF para Ipv6.

Cuando un router aprende múltiples rutas para una red especifica a través de múltiples procesos de enrutamiento (o protocolos de enrutamiento), este captura la ruta que posee la menor distancia administrativa dentro de la tabla de enrutamiento. En algunos casos el router debe seleccionar una ruta de muchas que han sido aprendidas a través del mismo proceso de enrutamiento y que poseen la misma distancia administrativa. En este caso el router elige la ruta con el menor costo o métrica hacia
el destino. Cada proceso de enrutamiento calcula un costo de manera distinta y en algunas ocasiones el costo necesita ser manipulado para lograr balancear la carga.

OSPF realiza el balance de carga automáticamente de la siguiente manera. Si OSPF encuentra una manera de alcanzar un destino a través de más de una interfase y cada ruta posee el mismo costo, este captura cada ruta en la tabla de enrutamiento. Para restringir el numero de rutas que van hacia un mismo destino se utiliza el comando **maximum-paths.** Por defecto el mayor número de rutas es 16; pero puede ser configurada de 1 a 64 rutas.

Implementando OSPF para Ipv6.

Habilitando OSPF en una interfase Ipv6.

Los siguientes pasos describen la forma de habilitar OSPF para el enrutamiento Ipv6 y configurarlo sobre cada interfase. Por defecto, el enrutamiento OSPF para Ipv6 se encuentra deshabilitado y mucho menos se encuentra configurado para las interfases.

A continuación se presentan los pasos para habilitar OSPF, de forma detallada.

PASO	COMANDO	PROPOSITO
1	**enable** **Ejemplo:** Router> enable	Habilita el modo EXEC privilegiado.
2	**configure terminal** **Ejemplo:** Router# configure terminal	Accesa al modo de configuración global.
3	**interface** interface-type interface-number **Ejemplo:** Router(config)# interface ethernet 0/0	Especifica el número, el tipo de interface y entra al modo de configuración de interface.
4	**ipv6 ospf** process-id **area** area-id [**instance** instance-id] **Ejemplo:** Router(config-if)# ipv6 ospf 1 area 0	Habilita OSPF para Ipv6 sobre una interfase.

Tabla 18 Habilitando OSPF en una Interfase.

Definiendo un área OSPF para Ipv6.

El costo sumarizado de un conjunto de rutas, será el costo mayor de las rutas que están siendo sumarizadas; por ejemplo, si las siguientes rutas son sumariadas:

```
OI 2003:0:0:7::/64 [110/20]
        via FE80::A8BB:CCFF:FE00:6F00, Ethernet0/0
OI 2003:0:0:8::/64 [110/100]
        via FE80::A8BB:CCFF:FE00:6F00, Ethernet0/0
OI 2003:0:0:9::/64 [110/20]
        via FE80::A8BB:CCFF:FE00:6F00, Ethernet0/0
```

La ruta sumarizada resultante seria la siguiente:

OI 2003::/48 [110/100]

via FE80::A8BB:CCFF:FE00:6F00, Ethernet0/0

Para establecer un área OSPF es necesario sumarizar las rutas pertenecientes a dicha área. Para lograr esto se deben seguir los siguientes pasos.

A continuación se presentan los pasos en forma detallada.

PASO	COMANDO	PROPOSITO
1	**enable** **Ejemplo:** Router> enable	Habilita el modo EXEC privilegiado.
2	**configure terminal** **Ejemplo:** Router# configure terminal	Accesa al modo de configuración global.
3	**ipv6 router ospf** process-id **Ejempo:** Router(config)# ipv6 router ospf	Habilita el modo de configuración de OSPF en el router.
4	**area** area-id **range** { ipv6-prefix/ prefix-length} [**advertise** | **not-advertise**] [**cost** cost] **Ejemplo:** Router(config-rtr)# area 1 range 2001::/48	Sumariza las rutas en una sola área.

Tabla 19 Definiendo un área OSPF para Ipv6.

12.1.3 BGP para IPv6.

Cuando se configura el Multiprotocolo BGP sobre Ipv6, se debe crear primeramente un proceso de enrutamiento BGP, configurar relaciones con vecinos y personalizar BGP para una red en particular.

Configurando un Proceso de Enrutamiento BGP para Ipv6 y un BGP Router ID.

Antes de configurar el router para correr BGP sobre Ipv6, se debe habilitar el enrutamiento Ipv6 globalmente en el router, utilizando el comando **ipv6 unicast-routing** en el modo de configuración global.

BGP utiliza un Router ID (Identificador del router) para identificar a otros vecinos BGP. El Router ID es un valor de 32 bits que frecuentemente es representado por una dirección Ipv4. Por defecto, el software Cisco IOS establece como router ID a la dirección Ipv4 de la interfase loopback del router. Si una interfase loopback no se encuentra configurada en el router, entonces se elige la dirección Ipv4 más alta configurada sobre las interfases del router para representar el Router ID.

Cuando se configura BGP sobre un router que esta habilitado solamente para Ipv6, el router no posee una dirección Ipv4, por lo tanto el Router ID se debe configurar manualmente sobre el router. El Router ID debe ser único para los vecinos BGP del router.

A continuación se presentan estos pasos en forma detallada.

PASO	COMANDO	PROPOSITO
1	**enable** **Ejemplo:** Router> enable	Habilita el modo EXEC privilegiado.
2	**configure terminal** **Ejemplo:** Router# configure terminal	Accesa al modo de configuración global.

3	**router bgp** autonomous-system-number **Ejemplo:** Router(config)# router bgp 65000	Configura un Proceso de Enrutamiento BGP y entra al modo de configuración del router de acuerdo al proceso de enrutamiento especificado.
4	**no bgp default ipv4-unicast** **Ejemplo:** Router(config-router)# no bgp default ipv4-unicast	Deshabilita las familias de direcciones Unicast Ipv4 para el proceso de enrutamiento BGP especificado en el paso anterior.
5	**bgp router-id** ip-address **Ejemplo:** Router(config-router)# bgp router-id 172.16.10.1	Configura un router ID de 32 bits como identificador del router local corriendo BGP. Al utilizar el comando **bgp router-id** se resetean todas las sesiones activas con los vecinos.

Tabla 20 Implementando BGP para Ipv6.

Configurando un Vecino en BGP para Ipv6.

Por defecto, los vecinos que están definidos utilizando el comando **neighbor remote-as** en el modo de configuración del router, intercambian solamente prefijos de direcciones Ipv4. Para que exista un intercambio de prefijos de direcciones de otros tipos, como los prefijos Ipv6, los vecinos deben activar el comando **neighbor activate** en el modo de configuración de familias de direcciones para los otros tipos de prefijos.
Los siguientes pasos se utilizan para configurar un vecino BGP para Ipv6.

A continuación se muestran estos pasos en forma detallada.

PASO	COMANDO	PROPOSITO
1	**enable** **Ejemplo:** Router> enable	Habilita el modo EXEC privilegiado.
2	**configure terminal** **Ejemplo:**	Accesa al modo de configuración global.

	Router# configure terminal	
3	**router bgp** autonomous-system-number **Ejemplo:** Router(config)# router bgp 65000	Configura un Proceso de Enrutamiento BGP y entra al modo de configuración del router de acuerdo al proceso de enrutamiento especificado.
4	**neighbor** ipv6-address **remote-as** autonomous-system-number **Ejemplo:** Router(config-router)# neighbor 2001:1110::9 remote-as 64450	Agrega la dirección Ipv6 del vecino en el sistema autónomo especificado, a la tabla de vecinos de BGP Ipv6 en el router local.
5	**address-family ipv6** [**unicast** | **multicast**] **Ejemplo:** Router(config-router)# address-family ipv6	Especifica la Familia de Direcciones Ipv6 y entra al modo de configuración de la misma. La palabra Unicast especifica la familia de direcciones unicast Ipv6. Esta opción viene por defecto. La palabra multicast especifica los prefijos de direcciones multicast Ipv6.
6	**neighbor** ipv6-address **activate** **Ejemplo:** Router(config-router-af)# neighbor 2001:1110::9 activate	Habilita el intercambio de prefijos Ipv6 con el vecino.

Tabla 21 Configurando un Vecino en BGP para Ipv6.

12.2 Protocolos de enrutamiento en Linux (Demonio Quagga)

Para configurar los demonios Ripngd, OSPv3 y BGP se puede consultar la sección **10.5.2.1 Configuración de Linux con Zebra**. De la misma forma que se configura el archivo principal de zebra se hace para estos demonios, por ejemplo para configurar RIPngd, se toma la configuración básica del archivo **ripngd.sample**, y se coloca la información en un nuevo archivo con el nombre de **ripngd.conf,** siempre dentro del mismo directorio **/etc/quagga**.

12.2.1 Demonio de RIPngd

Ripngd soporta el protocolo RIPng, el cual esta descrito en la RFC2080. Este protocolo es la adaptación del protocolo RIP a IPv6.

Configuración de ripng

Actualmente ripngd soporta los siguientes comandos:

- Comando: **router ripng *{}***
  - Habilita RIPng.
- Comando de RIPng: **flush_timer *time {}***
  - Configura el tiempo de nivelado.
- Comando de RIPng: **network *network {}***
  - Configura el RIPng como habilitado en la red.
- Comando de RIPng: **network *ifname {}***
  - Configura el IPng como habilitado en el interfaz por el nombre del interfaz
- Comando de RIPng: **route *network {}***
  - Configura el anuncio de routing estático a la red por RIPng.
- Comando: **router zebra *{}***
  - Este comando está habilitado por defecto y no aparece en la configuración. Mediante este comando las rutas de RIPng van directamente al demonio de RIPng.

Comandos del Modo Terminal de ripng

- Comando: **show ip ripng *{}***
- Comando: **show debugging ripng *{}***
- Comando: **debug ripng events *{}***
- Comando: **debug ripng packet *{}***
- Comando: **debug ripng zebra {}**

12.2.2 Demonio de OSPv3

ospf6d es un demonio que soporta la versión de OSPF 3 para redes en IPv6. OSPF v3 está descrito en la RFC2740.

Router OSPF6

- Comando: **router ospf6 {}**
- Comando de OSPF6: **router-id *a.b.c.d {}***
    - Configura el ID del router.
- Comando de OSPF6: **interface ifname area *area {}***
    - Asigna el interfaz a un área específica, y empieza a mandar paquetes OSPF. Area puede ser específicado con 0.

Área OSPF6

El soporte de área de OSPFv3 todavía no está implementado.

Interfaz OSPF6

- Comando de Interfaz: **ipv6 ospf6 cost COSTE *{}***
    - Configura el coste de salida del interfaz. El valor por defecto es 1.
- Comando de Interfaz: **ipv6 ospf6 hello-interval INTERVALOHELLO *{}***
    - Configura el intervalo de envío de mensajes Hello, el valor por defecto es 40
- Comando de Interfaz: **ipv6 ospf6 dead-interval INTERVALODEAD *{}***
    - Configura el intevalo de muerte del interfaz del router, por defecto este valor es de 40.
- Comando de Interfaz: **ipv6 ospf6 retransmit-interval INTERVALO DE RETRASMISION *{}***

    - Configura el intervalo de retransmisión. Por defecto es 5.
- Comando de Interfaz: **ipv6 ospf6 priority PRIORIDAD *{}***
    - Configura la prioridad del interfaz del router. Por defecto es 1.
- Comando de Interfaz: **ipv6 ospf6 transmit-delay RETARDORETRANSMISION *{}***
    - Configura el Inf-Trans-Delay. El valor por defecto es 1.

Redistribución de rutas a OSPF6

- Comando de OSPF6: **redistribute static *{}***
- Comando de OSPF6: **redistribute connected *{}***
- Comando de OSPF6: **redistribute ripng *{}***

Mostrar la Información de OSPF6

- Comando: **show ipv6 ospf6 [INSTANCIA_ID] *{}***
    - INSTANCIA_ID es un ID opcional de instancia de OSPF. Para ver el router ID y la instancia ID, hay que escribir "show ipv6 ospf6".
- Comando: **show ipv6 ospf6 database *{}***
    - Este comando muestra la base de datos de LSA. Es posible especificar el tipo de LSA.
- Comando: **show ipv6 ospf6 interface *{}***
    - Para ver la configuración de los interfaces de OSPF como costes.
- Comando: **show ipv6 ospf6 neighbor *{}***
    - Muestra el estado y el BDR elegido.
- Comando: **show ipv6 ospf6 request-list A.B.C.D *{}***
    - Muestra la lista requerido de vecinos.
- Comando: **show ipv6 route ospf6 *{}***
    - Este comando muestra la información interna de la tabla de routing

12.2.3 Demonio de BGP-4

Bgpd es un demonio de BGP-4 (Border Gateway Protocol 4, Protocolo de Pasarela Fronteriza 4). BGP-4 se describe en el RFC1771. bgpd también soporta las Extensiones Multiprotocolo para BGP-4 (también conocidas como BGP-4+ o MBGP) que se describen en el RFC2283.

BGP-4 es uno de los EGPs (Exterior Gateway Protocols, Protocolos de Pasarela Exterior) y se para el encaminamiento entre dominios.

Router BGP

Lo primero que se debe hacer es activar el encaminamiento BGP con el comando *router bgp*. Para configurar el encaminamiento BGP, necesitas un número de SA (AS number). El número de SA es una identificación de sistema autónomo. El protocolo BGP usa el número de SA para detectar si la conexión BGP es interna o externa.

El número de SA es un número entero entre 1 y 65535. Como usar un número se describe en el RFC1930. Los números de SA del 64512 al 65535 se definen como números de uso privado. Los SA privados no deben nunca anunciarse a la Internet global.

- Comando: **router bgp NÚMERO-SA *{}***
    - Habilita el proceso de protocolo BGP con el número de SA especificado. Después de este comando, puedes introducir cualquier comando BGP. No pueden crearse un proceso BGP bajo un SA diferente a menos que se especifique Multi-instancia
- Comando: **no router bgp NÚMERO-SA *{}***
    - Destruye un proceso BGP con el NÚMERO-SA especificado.
- Comando BGP: **bgp router-id ROUTER-ID *{}***
    - Este commando especifica el router-ID (Indetificador de router). Si bgpd conecta con zebra, obtiene la información de los interfaces y de dirección. En ese caso el router-id por defecto se obtiene tomando la dirección de mayor numeración de todos los interfaces. Cuando router zebra no está habilitado, bgpd no puede obtener la información de los interfaces y router-id se fija en 0.0.0.0. En ese caso se debe especificar a mano.

Distancia BGP

- Comando BGP: **distance bgp <1-255> <1-255> <1-255>*{}***
    - Este comando cambia el valor de la distancia de BGP. Cada argumento es un valor de distancia para rutas externas, rutas internas y rutas locales.
- Comando BGP: **distance <1-255> A.B.C.D/M *{}***
- Comando BGP: **distance <1-255> A.B.C.D/M word *{}***
    - Este comando configura el valor de la distancia a un desino.

Proceso de decisión de BGP

- Comprobación de pesos.
- Comprobación de preferencia local.
- Comprobación de ruta local.
- Comprobación de longitud del camino del Sistema Autónomo.
- Comprobación del origen.
- Comprobación de MED.

Red BGP

Ruta BGP

- Comando BGP: **network *A.B.C.D/M {}***
    - Este comando activa el anuncio de esa red.

router bgp 1
network 10.0.0.0/8

Esta configuración de ejemplo nos dice que la red 10.0.0.0/8 será anunciada a todos los vecinos. Varios vendedores de routers no anuncian las rutas si estas no están presentes en sus tablas de encaminamiento IGP; *bgp* no tiene en cuenta su las rutas están anunciadas en las rutas IGP·

- Comando BGP: **no network *A.B.C.D/M {}***
    - Desactiva el anuncio de prefijo previamente anunciado.

Agregación de Rutas

- Comando BGP: **aggregate-address *A.B.C.D/M {}***
    - Este comando especifica que se agregen un conjunto de rutas recibidas en un PREFIJO menos especifico.
- Comando BGP: **no aggregate-address *A.B.C.D/M {}***
    - Desactiva la agregación de prefijos.
- Comando BGP: **aggregate-address A.B.C.D/M summary-only *{}***
    - Este comando especifica una dirección agregada. Las rutas agregadas no serán anunciadas
- Comando BGP: **no aggregate-address A.B.C.D/M *{}***

Redistribución a BGP

- Comando BGP: **redistribute kernel *{}***
    - Inyecta las rutas del kernel que no pertenecen a zebra en el proceso de BGP.
- Comando BGP: **redistribute static *{}***
    - Inyecta las rutas estáticas de zebra en el proceso de BGP.
- Comando BGP: **redistribute connected *{}***
    - Inyecta las rutas conectadas de zebra en el proceso de BGP.
- Comando BGP: **redistribute rip *{}***
    - Inyecta las rutas de ripd en el proceso BGP.
- Comando BGP: **redistribute ospf *{}***
    - Inyecta las rutas de ospfd en el proceso de BGP.

Vecino BGP

- Comando BGP: **neighbor *VECINO* remote-as *NÚMERO-SA {}***
    - Crea un nuevo vecino cuyo SA es NÚMERO-SA. VECINO puede ser una dirección IPv4 ó IPv6.
- Comando BGP: **no neighbor *VECINO* remote-as *NÚMERO-SA {}***
    - Elimina un vecino y toda la configuración asociada a él.

Configuración de vecinos

- Comando BGP: neighbor *VECINO* shutdown *{}*
- Comando BGP: no neighbor *VECINO* shutdown *{}*
    - Desactiva el vecino manteniendo la configuración asociada a este. Podemos desactivar un vecino con "no neighbor VECINO remote-as NÚMERO-SA pero a la vez borraremos la configuración asociada. Usa esta sintaxis cuando quieras tirar la sesión con un vecino pero preservando toda su configuración. Con la operación "no neighbor VECINO shutdown" activaremos la negociación de nuevo.
- Comando de BGP: **neighbor *VECINO* ebgp-multihop [TTL] *{}***
- Comando de BGP: **no neighbor *VECINO* ebgp-multihop *{}***
    - Activa el modo ebgp-multihop, usado para establecer sesisiones BGP entre sistemas de distintos SA, con no se encuentran directamente conectados al mismo segmento de red y en ciertas configuraciones de tunnel, gre e ipip. TTL establece el número de saltos al los que se encuentra el vecino, si no se especifica, el valor por defecto es el máximo, 255. Con "no" se desactiva la configuración ebgp-multihop.
- Comando de BGP: **neighbor *VECINO* description *{}***
- Comando de BGP: **no neighbor *VECINO* description *{}***
    - Establece/Elimina una descripción del vecino en texto libre.
- Comando de BGP: **neighbor *VECINO* version *VERSION* *{}***
- Comando de BGP: **no neighbor *VECINO* version *VERSION* *{}***
    - Fija la versión BGP del vecino que puede ser 4, 4+ o 4-. BGP version 4, es el valor por defecto para conecxiones BGP. BGP version 4+ significa que el vecino soporta Extensiones Multiprotocolo para BGP-4. BGP version 4- es similar pero el vecino habla Extensiones Multiprotocolo para BGP-4 en la antigua versión "Internet-Draft revision 00". Algún software de enrutamiento todavía lo usa.
- Comando de BGP: **neighbor *VECINO* interface *NOMBRE_IF* *{}***
- Comando de BGP: **no neighbor *VECINO* interface *NOMBRE_IF* *{}***
    - Cuando conecta con un vecino BGP sobre una dirección IPv6 de enlace-local, debes especificar el nombre del interfaz usado para esa conexión.
- Comando BGP: **neighbor *VECINO* next-hop-self *{}***
- Comando BGP: **no neighbor *VECINO* next-hop-self *{}***

    - Este comando fuerza al encaminador a anunciarse como el próximo salto, para las rutas que distribuya a sus vecinos.
- Comando BGP: **neighbor *VECINO* update-source *NOMBRE_IF {}***
- Comando BGP: **no neighbor *VECINO* update-source *NOMBRE_IF {}***
    - Con este comando, bgpd usará la dirección del interfaz designado para establecer la sesión BGP, es casi imprescindible cuando se usa ebgp-multihop.
- Comando BGP: **neighbor *VECINO* default-originate *{}***
- Comando BGP: **no neighbor *VECINO* default-originate *{}***
    - El comportamiento por defecto de bgpd es no anunciar la ruta por defecto (0.0.0.0/0), incluso si esta se encuentra en la tabla de rutas. Este comando anunciará la ruta (0.0.0.0/0) a ese vecino concreto, si existe, o le generará una.
- Comando BGP: **neighbor *VECINO* port *PUERTO {}***
- Comando BGP: **no neighbor *VECINO* port *PUERTO {}***
    - Especifica que puerto tcp se usará para establecer la sesión BGP con ese vecino si es distinto del puerto estándar (179).
- Comando BGP: **neighbor *VECINO* send-community *{}***
- Comando BGP: **no neighbor *VECINO* send-community *{}***
    - Instruye al demonio BGP a enviar las comunidades, asociadas a los distintos prefijos, este el comportamiento por defecto de bgpd, por lo que, debemos instruirle "no neighbor VECINO send-community" para deshabilitar el envío de comunidades.
- Comando BGP: **neighbor *VECINO* weight *PESO {}***
- Comando BGP: **no neighbor *VECINO* weight *PESO {}***
    - Este comando especifica un peso por defecto a las rutas aprendidas de ese vecino. (Es específico de Cisco).
- Comando BGP: **neighbor *VECINO* maximum-prefix *NÚMERO {}***
- Comando BGP: **no neighbor *VECINO* maximum-prefix *NÚMERO {}***
    - Este comando fija un tope máximo de prefijos que un vecino nos puede enviar, y se usa para evitar una inundación de prefijos que ponga en peligro la estabilidad de la red. Al llegar al número máximo de prefijos la sesión se cerrará hasta que el administrador, ejecute el comando "no neighbor VECINO shutdown"

Capítulo

13

Nat en IPv6

Temas:

- NAT.
- NAT en Ip6tables.
- NAT-PT.

13. NAT en IPv6

La utilización de una técnica conocida como traducción de direcciones de red (NAT) ha llevado a no dar importancia a la escasez de espacio de direccionamiento IPv4. Si bien la técnica NAT permite que muchos dispositivos interconectados posean su propia dirección local, éstos se conectan a Internet a través de un dispositivo con una sola dirección IPv4.

Por consiguiente, aunque el usuario (cliente) de un dispositivo NAT pueda comunicarse con servidores de Internet en el marco del modelo de comunicaciones "cliente-servidor", no se podrá garantizar la accesibilidad a dicho usuario (cliente), si los usuarios de dispositivos externos desean establecer conexiones. Debido a este planteamiento en IPv6 se menciona que: "La tecnología NAT supone la quiebra del principio de extremo a extremo de Internet y por ello impide pasar a la siguiente generación de aplicaciones, las cuales requieren espacio y direcciones IP, y obligan a ofrecer conectividad a las redes en los locales de las empresas y los hogares (por ejemplo, a partir de aparatos móviles basados en IP)".

13.1 Nat en Ip6tables

El comando utilizado para manipular el filtrado de red de IPv6 es ip6tables. La mayoría de las directivas para este comando son idénticas a aquellas usadas por iptables, **excepto que la tabla nat aún no es compatible**. Esto significa que todavía no es posible realizar tareas de traducción de direcciones de red IPv6, tales como enmascarado y reenvío de puertos.

NAT para redes nativas IPv6 no se puede implementar todavía. Sin embargo, actualmente las redes se encuentran mezcladas con los protocolo ipv4 e ipv6 y para manejar las traducciones de direcciones en ambos protocolos, existe un protocolo que se llama NAT-PT, el cual se explica a continuación.

13.2 NAT-PT

IPv6 es un nuevo protocolo de IETF que trata de solucionar la actual falta de direcciones IPv4 y limitar el crecimiento de las tablas de *routing* en el núcleo de Internet. Además, IPv6 incorpora muchas funcionalidades de manera nativa que IPv4 ha ido incorporando mediante parches sucesivos (IPsec, Movilidad, etc.).

Para no afectar los servicios que proporcionan las actuales infraestructuras IPv4, el despliegue de la nueva infraestructura IPv6 se ha hecho de una manera progresiva. La primera fase del despliegue es naturalmente una fase de transición en la que IPv4 e IPv6 deben de coexistir. Para esta fase se requieren *Mecanismos de Transición* que permitan la interoperación IPv4 e IPv6.

Existen multitud de *Mecanismos de Transición*, que proporcionan dicha interoperabilidad, pero uno de los más flexibles y potentes es NAT-PT (*Network Address Translation - Protocol Translation*). NAT-PT esta basado en el NAT de IPv4, solo que NAT-PT no se limita a cambiar la dirección, sino que traduce completamente cabeceras de IPv4 a IPv6 y viceversa.

Existen diferentes implementaciones de NAT-PT, una de las más conocidas es la del *kernel* experimental para FreeBSD KAME <http:/ /www.kame.net>, sin embargo el dato importante es que actualmente CISCO ya lo incorpora en su sistema operativo (IOS, *Internetworking Operating System*) como una funcionalidad más. Este hecho indica que NAT-PT es uno de los mecanismos más importantes en la fase de transición IPv4-IPv6.

Implementaciones

BT Ultima

Ultima es una red experimental. Donde los dispositivos fueron habilitados para traducir direcciones IPv6 a IPv4 (y viceversa) y también se habilitó un túnel IPv6 sobre Ipv4. La traducción es con el mecanismo NAT-PT. El mecanismo Túnel, permite que cada túnel Ipv6 sobre IPv4 pueda ser autoconfigurado por medio de las peticiones DNS y las respuestas son generadas por el túnel broker (ver anexo). Ver figura 96.

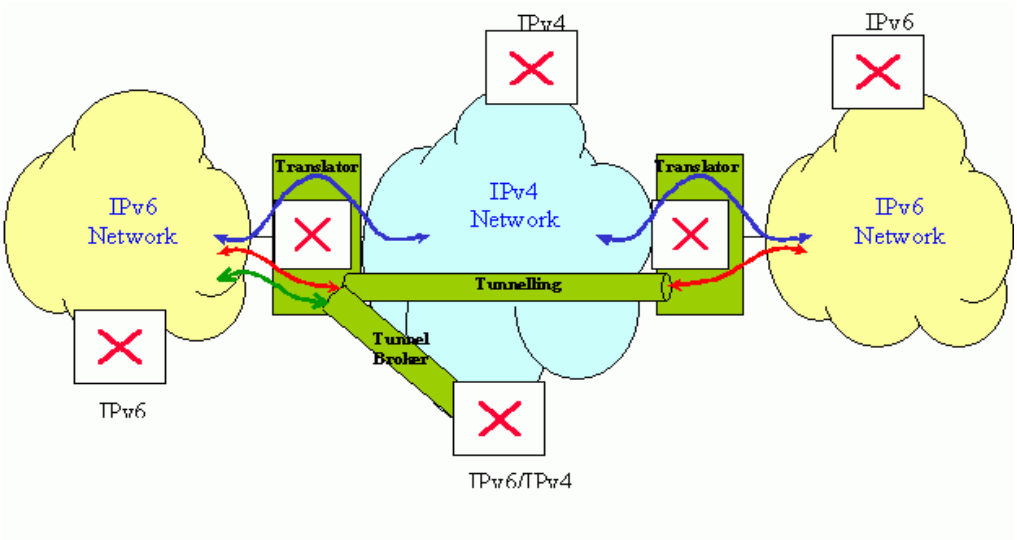


Figura 97. Topología de la red ultima.

13.2.1 Cisco NAT-PT

El protocolo traductor es distribuido como parte del IOS de CISCO en las implementaciones para IPv6. Actualmente las aplicaciones son experimentales. Existen dos implementaciones uno basado en la versión del IOS 11.3 y otro basado en la versión 12.0.

La traducción es bidireccional, NAT-PT permite el acceso desde ambas redes ipv6 a servidores IPv4, y desde redes IPv4 a servidores IPv6. Esta traducción TCP/UDP, ICMP y FTP lo realice la capa superior del protocolo de traducción. Este proceso se trabaja en conjunto con NAT para IPv4 y se utiliza el parámetro de carga útil de IPv4, esto es posible usando una dirección IPv4 para una subred de una dirección IPv6 (NAT-PT). Además es posible configurar mapeo estático entre direcciones IPv4 e IPv6.

Ejemplo de configuración para la traducción de IPv6 a IPv4:

```
IP NAT Pool IPv4Pool 192.168.2.0 192.168.2.255 PREFIX-LENGTH 24

IP NAT INSIDE SOURCE LIST 1 POOL IPv4Pool OVERLOAD

!

IPV6 NAT MAP V6V4 NET FEC::2:0/112 192.168.0.0
```

Una traducción de IPv6 a IPv4 trabaja de la siguiente manera:

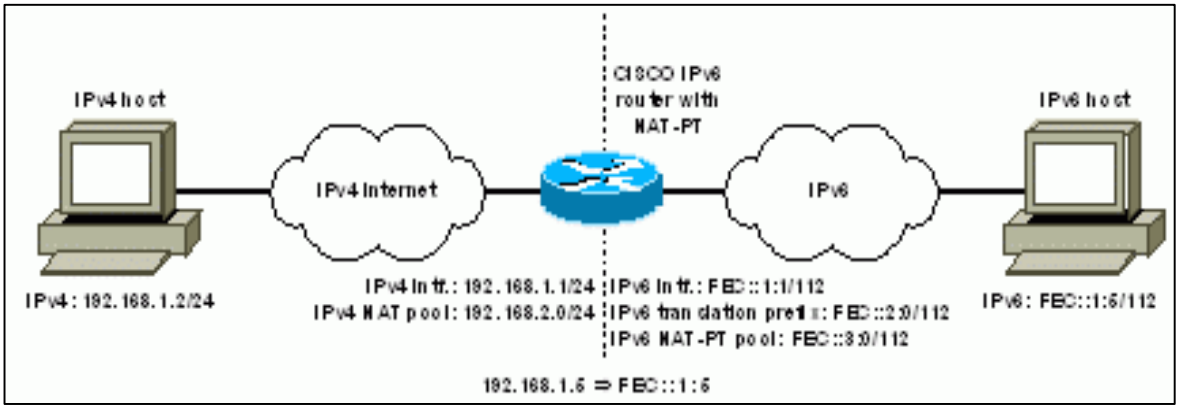


Figura 98. Implementación NAT-PT en CISCO

Cuando el host IPv6 desea comunicarse con el host IPv4, éste le envía paquetes con dirección fuente FEC::1:5 hacia el destino FEC::2:102. El router concatena los últimos 16 bits de la dirección destino (102) con el prefijo conocido de IPv4 que es 192.168.x.x de la forma de dirección IPv4 192.168.1.2. Entonces esto convierte la cabecera IPv6 a una cabecera IPv4 equivalente.

Recomendaciones

El protocolo IPv6 es una realidad, no es un proyecto que se espera que se implemente en el futuro. La mayoría de los sistemas operativos como Windows, Linux, Mac, y dispositivos de red, por ejemplo CISCO, 3COM, soportan el protocolo ipv6, por lo tanto, se recomienda:

1. La capacitación y el estudio para estar preparado ante este cambio de protocolo. La capacitación debe de ser estándar es decir, desde un Jefe Administrador de Red en una empresa hasta un estudiante de una universidad. Por que el cambio nos involucra a todos.

2. Debido que el protocolo funciona correctamente en software GNU, se pueden realizar más trabajos de investigación e implementación del protocolo IPv6, como por ejemplo:

- Creación de DNS con el demonio BIND.
- Implementación del demonio RADVD, que se utiliza para la autoconfiguración de direcciones en una red.
- Además también es importante conocer como se manejan Linux la configuración de los distintos túneles para empaquetar ipv6 en ipv4.

3. Muchas Universidades y organizaciones extranjeras, se esfuerzan para investigar el protocolo ipv6, pero la mayoría de esta información se encuentran en inglés, o los casos de estudios son muy complejos y no muy entendibles. Ante esta limitante se recomienda, que se debería de fomentar la investigación en nuestro país para formar un grupo y compartir experiencias.

4. Como recomendación final, si se desea implementar el protocolo ipv6 en una red, en las computadoras, se recomienda configurar Windows XP, o Windows Vista, por que en versiones anteriores, la pila del protocolo ipv6 no funciona muy bien, debido a que el protocolo no es parte del sistema operativo, si no que, hay que agregarlo como una herramienta extra. Para el caso de Linux, cualquier distribución es recomendable, solo si el kernel es 2.6 o superior. Y en equipos de red como CISCO, la versión del IOS es muy importante, se recomienda a partir de la 12.2 (2)T.

Conclusiones

- Se concluye que el protocolo ipv6 es estándar, es decir que se puede configurar fácilmente en cualquier sistema operativo y en cualquier equipo de red. La mayoría de los sistemas operativos tienen incorporado en la pila TCP/IP este protocolo, por lo tanto solo se necesita saber configurarlo para poder hacer pruebas.

- Una de las características importantes del protocolo IPv6 es la autoconfiguración, es decir que en el router se ejecuta el módulo y automáticamente se puede configuran las direcciones ipv6 en los clientes. Además otra característica principal y gran diferencia con el protocolo ipv4 es el formato de las direcciones. Es necesario saber identificar las direcciones para conocer el rango que tiene, por ejemplo si es local o global.

- Gracias a la cantidad de direcciones que se obtiene con el protocolo ipv6, una tarjeta de red puede tener configurada muchas direcciones ipv6 y funcionan correctamente. La configuración básica del protocolo es similar a la del ipv4, es decir que se necesita de una dirección ip y una dirección de puerta de enlace.

- Linux posee herramientas potentes como las de un enrutador sofisticado, y es bueno aprovechar estas características, debido al costo que implica obtener enrutadores de marca. En Linux existe el demonio Quagga, el cual posee muchos rasgos similares a la configuración en equipos CISCO.

- A parte de enrutador Linux puede funcionar como un firewall, en este caso se emplea la herramienta netfilter6 por medio de ip6tables, las cuales se configuran de una forma similar a iptables, con la diferencia que no tienen soporte para NAT, debido a que con IPv6 no se necesita de NAT, por la cantidad de direcciones que se pueden formar.

FUENTES DE INFORMACION

BIBLIOGRAFÍA:

1. ESCALANTE RENDÓN, JOSÉ ISAAC. **GUÍA DE IMPLEMENTACIÓN DE REDES UTILIZANDO EL PROTOCOLO IPV6**. Universidad Don Bosco, 2005.
2. JOSEPH DAVIES, **UNDERSTANDING IPV6,** Microsoft Corporation 2003. ISBN 0-7356-1245-5.
3. SILVIA HAGEN, **IPV6 ESSENTIALS,** Primera Edición, julio 2002 .ISBN: 0-596-00125-8
4. ILJITSCH VAN BEIJNUM , **RUNNING IPV6**, Apress, ISBN: 1-59059-527-0
5. SAM BROWN, **CONFIGURING IPV6 FOR CISCO IOS.** Syngress .ISBN: 1-928994-84-9

SITIOS DE INTERNET:

1. **http://www.6sos.net/**

   Brinda información sobre la instalación y configuración en los distintos sitemas operativos.

2. **http://www.IPv6day.org/action.php?n=Es.IPv6day**

   IPv6 day, nuevo sitio, en lugar de 6Bone.

3. **http://www.bieringer.de/linux/IPv6/**

   Sitio con información del manejo de ipv6 en Linux.

4. **http://www.rfc-editor.org/rfcxx00.html**

   Página oficial de Internet Protocols, RFC.

5. **http://www.IPv6style.jp/en/index.shtml**

   Sitio web de Japón que funciona con ipv6, tiene documentos de ayuda traducidos en inglés.

6. **http://redes-linux.all-inone.net/manuales/IPv6/Tutorial_de_IPV6.pdf**

   Este sitio ofrece tutoriales de IPv6, proporcionado por RedesLinux.com.

7. **http://www.eduangi.com/quagga/.**

   Manual del demonio Quagga para Linux.

8. **http://codarec.frm.utn.edu.ar/areas/firewall6//manip6tables//man-ip6tables.html**

   En esta página se encuentra una explicación de los comandos de ip6tables.

9. **http://www.consulintel.es/**

   Consultoría de Telecomunicaciones y miembro fundador de Ipv6 Forum.

GLOSARIO

- **AES**

Advanced Encryption Standard (AES), también conocido como Rijndael, es un esquema de cifrado adoptado como un estándar de encriptación por el gobierno de los Estados Unidos, como también analizado exhaustivamente, como fue el caso de su predecesor, el Estándar de Encriptación de Datos (DES). Fue adoptado por el Instituto Nacional de Estándares y Tecnología (NIST).

- **ACL**

Listas de Control de Acceso (en inglés Access Control Lists, o ACL) permiten controlar el flujo del tráfico en equipos de redes, tales como routers y switches. Su principal objetivo es filtrar tráfico, permitiendo o negando el tráfico de red de acuerdo a alguna condición.

- **AH**

(Authentication Header) Cabecera de autenticación AH; asegura la autenticidad y la integridad de los datos incluyendo campos invariantes.

- **ALICE**

(América Latina Interconectada Con Europa) se ha establecido para crear una infraestructura de redes de investigación utilizando el protocolo IP dentro de América Latina y su interconexión con Europa.

- **ATM**

Modo de Transferencia Asíncrona o Asynchronous Transfer Mode (ATM) es una tecnología de telecomunicación desarrollada para hacer frente a la gran demanda de capacidad de transmisión para servicios y aplicaciones.

- **DATAGRAMA**

Agrupación lógica de información que se envía como una unidad de capa de red a través de un medio de transmisión sin establecer con anterioridad un circuito virtual. Los datagramas IP son las unidades principales de información de Internet.

- **DESENCRIPTAR**

Decodificar los datos para obtener la información en claro

- **DHCP**

Son las siglas en inglés de Protocolo de configuración dinámica de servidores (Dynamic Host Configuration Protocol). Es un protocolo de red en el que un servidor provee los parámetros de configuración a las computadoras conectadas a la red informática que los

requieran (máscara, puerta de enlace y otros) y también incluye un mecanismo de asignación de direcciones de IP.

- **ENCRIPTAR**

Codificar los datos en función a una clave.

- **ESP**

Encapsulación de la Carga de Seguridad, es un protocolo de autenticación y cifrado que usa mecanismos criptográficos para proporcionar integridad, autenticación del origen, y confidencialidad.

- **EXTRANET**

Red externa a una organización. No propiedad de la organización.

- **FIREWALLS**

Es un elemento de hardware o software utilizado en una red de computadoras para prevenir algunos tipos de comunicaciones prohibidos según las políticas de red que se hayan definido en función de las necesidades de la organización responsable de la red. Su modo de funcionar es definido por la recomendación RFC 2979 [84], la cual define las características de comportamiento y requerimientos de interoperabilidad.

- **FRAME RELAY**

Es una técnica de transmisión a conmutación de paquetes, introducida por la ITU-T a partir de la recomendación I.122 de 1988. Consiste en una forma simplificada de tecnología de conmutación de paquetes que transmite una variedad de tamaños de marcos ("frames") para datos, perfecto para la transmisión de grandes cantidades de datos.

- **FTP**

Protocolo de transferencia de archivos. Protocolo de aplicación, parte de la pila de protocolo TCP/IP, que se usa para transferir archivos entre nodos de la red.

- **HACKERS**

Es el neologismo utilizado para referirse a un experto en varias o alguna rama técnica relacionada con las tecnologías de la información y las telecomunicaciones: programación, redes de computadoras, sistemas operativos, hardware de red/voz, etc.

- **HASH**

Algoritmo que trasforma los datos en un resumen irreversible. Es decir, si pasamos los datos por un algoritmo Hash obtendremos un resumen único de esos datos, pero a partir del resumen no seremos capaces de volver a obtener los datos.

---

[84] *RFC 2979 "Comportamiento y requerimientos para Firewalls."*

- **ICMP**

(Internet Control Message Protocol); es el subprotocolo de diagnóstico y notificación de errores del Protocolo de Internet (IP). Como tal, se usa para enviar mensajes de error, indicando por ejemplo que un servicio determinado no está disponible o que un router o host no puede ser localizado.

- **IETF**

Fuerza de Tareas de Ingeniería de Internet, compuesta por alrededor de 80 grupos de trabajo que tienen la responsabilidad de desarrollar estándares de Internet.

- **INTERNET**

Es una red de redes a escala mundial de millones de computadoras interconectadas con un conjunto de protocolos, el más destacado, el TCP/IP. También se usa este nombre como sustantivo común y por tanto en minúsculas para designar a cualquier red de redes que use las mismas tecnologías que Internet, independientemente de su extensión o de que sea pública o privada.

- **INTRANET**

Red local propiedad de una organización.

- **ISAKMP**

IPSec usa el protocolo Asociación de seguridad en Internet y administración de claves (ISAKMP, Internet Security Association and Key Management Protocol) para intercambiar y administrar dinámicamente claves cifradas entre los equipos que se comunican.

- **IP**

Protocolo Internet. Protocolo de capa de red en la pila TCP/IP que brinda un servicio de internetworking no orientado a conexión. El IP suministra características de direccionamiento, especificación de tipo de servicio, fragmentación y reensamblaje y seguridad. Documentado en la RFC 791 [85].

- **IPSEC**

Internet Protocol security; es una extensión al protocolo IP que añade cifrado fuerte para permitir servicios de autenticación y cifrado y, de esta manera, asegurar las comunicaciones a través de dicho protocolo. Inicialmente fue desarrollado para usarse con el nuevo estándar IPv6, aunque posteriormente se adaptó a IPv4.

- **ISP**

Un proveedor de servicios de Internet (Internet Service Provider) es una empresa dedicada a conectar a Internet la línea telefónica de los usuarios o las distintas redes que

[85] *RFC 791, "IP Internet Protocol."*

tengan, y dar el mantenimiento necesario para que el acceso funcione correctamente. También ofrecen servicios relacionados, como alojamiento web o registro de dominios entre otros.

- **LAN**

Red de área local. Redes de datos de alta velocidad y bajo nivel de errores que abarcan un área geográfica relativamente pequeña (hasta unos pocos miles de metros). Las LAN conectan estaciones de trabajo, dispositivos periféricos, terminales y otros dispositivos que se encuentran en un mismo edificio u otras áreas geográficas limitadas.

- **MAC**

Control de acceso al medio. La más baja de las dos subcapas de la capa de enlace de datos definida por el IEEE. La subcapa MAC administra acceso al medio compartido como, por ejemplo, si se debe usar transmisión de tokens o contención.

- **MD5**

Message-Digest Algorithm 5, (Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits ampliamente usado. El código MD5 fue diseñado por Ronald Rivest en 1991, pero en un futuro cercano se cambiará de este sistema a otro más seguro.

- **MTU**

La unidad máxima de transferencia (Maximum Transfer Unit) es un término de redes de computadoras que expresa el tamaño en bytes del datagrama más grande que puede pasar por una capa de un protocolo de comunicaciones.

- **NAT**

Network Address Translation (Traducción de Dirección de Red) es un estándar creado por la Internet Engineering Task Force (IETF) el cual utiliza una o más direcciones IP para conectar varias computadoras a otra red (normalmente a Internet), los cuales tiene una dirección IP completamente distinta. Por lo tanto, se puede utilizar para dar salida a redes públicas que se encuentran con direccionamiento privado o para proteger máquinas públicas.

- **PAQUETE**

Un paquete de datos es una unidad fundamental de transporte de información en todas las redes

- **P2P**

Peer-to-peer es una red informática entre iguales, se refiere a una red que no tiene clientes y servidores fijos, sino una serie de nodos que se comportan simultáneamente

como clientes y como servidores de los demás nodos de la red, en donde el cliente y el servidor no pueden cambiar de roles.

- **PDA**

Personal Digital Assistant, (Ayudante personal digital), es un computador de mano originalmente diseñado como agenda electrónica. Hoy en día se puede usar como una computadora doméstica (ver películas, crear documentos, navegar por Internet...).

- **QoS**

(Quality of Service, Calidad de Servicio) Medida de desempeño para un sistema de transmisión que refleja su calidad de transmisión y disponibilidad de servicio.

- **RAICES**

RAICES es la Red Nacional de Investigación y Educación de El Salvador (NREN), es miembro fundador de CLARA (Cooperación Latinoamericana de Redes Avanzadas) y socio local de DANTE (Delivering Advanced Network To Europa) para el Proyecto ALICE (América Latina Interconecta con Europa).

- **RFC**

Conjunto de documentos que se usan como el medio principal para comunicar información acerca de Internet. La mayoría de las RFC documentan especificaciones de protocolo como, por ejemplo, TELNET y FTP, están disponibles en línea en varias fuentes. Las RFC's se utilizan como estándares de facto par a Internet.

- **ROAMING**

Roaming es un concepto utilizado en comunicaciones inalámbricas que está relacionado con la capacidad de un dispositivo para moverse de una zona de cobertura a otra.

- **ROUTER (ENRUTADOR)**

Dispositivo de capa de red que usa una o más métricas para determinar la ruta óptima a través de la cual se debe enviar el tráfico de red. Los routers envían paquetes desde una red a otra basándose en la información de la capa de red.

- **SCRIPT**

Es un guión o conjunto de instrucciones de un programa

- **SSH**

SSH (Secure SHell) es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo el ordenador mediante un intérprete de comandos.

- **SSL**

El protocolo SSL es un sistema diseñado y propuesto por Netscape Communications Corporation. Se encuentra en la pila OSI entre los niveles de TCP/IP y de los protocolos HTTP, FTP, SMTP, etc. Proporciona sus servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente con un algoritmo de cifrado simétrico, mediante un algoritmo de cifrado de clave pública, típicamente el RSA.

- **TCP**

Protocolo de control de transmisión. Protocolo de capa de transporte orientado a conexión que suministra transmisión de datos full-duplex confiable. El TCP forma parte de la pila de protocolo TCP/IP. Ver también *TCP/IP*.

- **TELNET**

Comando que se usa para verificar el software de capa de aplicación entre las estaciones origen y destino. Es el mecanismo de prueba más completo disponible. El puerto que se utiliza generalmente es el 23.

- **TTL**

(Time To Live), cuando se habla de Protocolo IP

- **TUNNELING**

Proceso por el que se crea una conexión VPN entre dos extremos a través de una red IP intermedia.

- **UDP**

Protocolo de Datagrama de Usuario. Protocolo de la capa de transporte no orientado a conexión de la pila de protocolos TCP/IP. El UDP es un protocolo simple que intercambia datagramas sin acuses de recibo ni garantía de envío, que requiere que el procesamiento de errores y la retransmisión sean administrados por otros protocolos. El UDP se define en la RFC 768 [86].

- **VoIP**

Voz sobre IP (VoIP) es, a grandes rasgos, un sistema de enrutamiento de conversaciones de voz mediante paquetes basados en el protocolo IP por la red de Internet.

- **VPN**

Es una red privada que se extiende, mediante un proceso de encapsulación y en su caso de encriptación, de los paquetes de datos a distintos puntos remotos mediante el uso de unas infraestructuras públicas de transporte.

[86] *Ver referencia en la Tabla 8 RFC's del Anexo, página 43*

- **WAN**

Red de área amplia. Red de comunicación de datos que sirve a usuarios dentro de un área geográficamente extensa y a menudo usa dispositivos de transmisión provistos por un servicio público de comunicaciones. Frame Relay, y X.25 son ejemplos de protocolos de comunicación de una red WAN.

- **X.25**

Red de conmutación de paquetes basada en el protocolo HDLC proveniente de IBM. Establece mecanismos de direccionamiento entre usuarios, negociación de características de comunicación, técnicas de recuperación de errores

- **X.509**

Es un estándar ITU-T para infraestructura de claves pública (Public Key Infrastructure o PKI). X.509 especifica, entre otras cosas, formatos estándar para certificados de claves públicas y un algoritmo de validación de ruta de certificación. X.509 es la pieza central de la infraestructura PKI, y es la estructura de datos que enlaza la clave pública con los datos que permiten identificar al titular.

Anexos

1. **Modelo OSI.**
2. **Ayuda de los comandos para la configuración de Ipv6 en Windows XP.**
3. **Tunnel Brokers.**

ANEXO 1 MODELO OSI

ISO (Organización Internacional de Estandarización). Es un organismo multinacional dedicado a establecer acuerdos mundiales sobre estándares internacionales. Un estándar ISO que cubre todos los aspectos de las redes de comunicación es el modelo de Interconexión de Sistemas Abiertos (OSI). Un Sistema Abierto es un modelo que permite que dos sistemas diferentes se puedan comunicar independientemente de la arquitectura subyacente. El modelo OSI permite la comunicación entre sistemas distintos sin que sea necesario cambiar la lógica del hardware o el software subyacente.

NIVEL FISICO: Coordina las funciones necesarias para transmitir el flujo de datos a través de un medio físico.

- Características físicas de las interfaces y el medio.
- Representación de los bits.
- Tasa de datos.
- Sincronización de los bits.
- Configuración de la línea.
- Topología física.
- Modo de Transmisión.

NIVEL DE ENLACE DE DATOS: Transforma el nivel físico, es un simple medio de transmisión en un enlace fiable y es responsable de la entrega nodo a nodo.
Hace que el nivel físico aparezca ante el nivel de red como un medio libre de errores. Divide el flujo de bits recibidos del nivel de red en unidades de datos manejables denominadas tramas.

- Tramado.
- Direccionamiento Físico.
- Control de Flujo.
- Control de Errores.
- Control de Acceso.

NIVEL DE RED: Responsable de la entrega de un paquete desde el origen al destino, y posiblemente a través de redes. Mientras el nivel de enlace de datos supervisa la entrega del paquete entre dos sistemas de la misma red, el nivel de red asegura que cada paquete va del origen al destino, sean estos cuales sean.

- Direccionamiento Lógico.
- Encaminamiento.

NIVEL DE TRANSPORTE: Es responsable de la entrega origen a destino de todo el mensaje. Mientras que el nivel de red supervisa la entrega extremo a extremo de paquetes individuales, no reconoce ninguna relación entre estos paquetes. El nivel de transporte asegura que todo el mensaje llega intacto y en orden. El nivel de red envía cada paquete a la computadora adecuada, el nivel de transporte envía el mensaje entero al proceso adecuado dentro de esa computadora.

- Direccionamiento en punto de servicio.
- Segmentación y reensamblado.
- Control de Conexión.
- Control de Flujo.

NIVEL DE SESION: Permite a usuarios de programas de máquinas diferentes establecer sesiones entre sí. Establece una sesión que requiere un proceso similar al de realizar una llamada telefónica: primero se llama al destinatario del mensaje y, si este contesta, se comienza a intercambiar información; en el caso del nivel de sesión, tendremos la sesión activa o establecida.

NIVEL DE PRESENTACION: Está relacionado con la sintaxis y semántica de la información intercambiada entre dos sistemas.

- Traducción
- Cifrado.
- Compresión.

NIVEL DE APLICACIÓN; Permite al usuario tanto humano como software acceder a la red. Proporciona las interfaces de usuario y soporte para servicios como el correo electrónico, el acceso y la transferencia de archivos remotos, la gestión de datos compartidos y otros tipos de servicios para información distribuida.

ANEXO 2 Configuración de Ipv6 en Windows XP .

1 Comandos IPV6

- ipv6 [-p] [-v] if [ifindex]
- ipv6 [-p] ifcr v6v4 v4src v4dst [nd] [pmld]
- ipv6 [-p] ifcr 6over4 v4src
- ipv6 [-p] ifc ifindex [forwards] [-forwards] [advertises] [-advertises] [mtu #bytes] [site id_sitio] [preference P]
- ipv6 rlu ifindex v4dst
- ipv6 [-p] ifd ifindex
- ipv6 [-p] adu ifindex/address [life validlifetime[/preflifetime]] [anycast] [unicast]
- ipv6 nc [ifindex [dirección]]
- ipv6 ncf [ifindex [dirección]]
- ipv6 rc [ifindex dirección]
- ipv6 rcf [ifindex [dirección]]
- ipv6 bc
- ipv6 [-p] [-v] rt
- ipv6 [-p] rtu prefix ifindex[/dirección] [life valid[/pref]] [preference P] [publish] [age] [spl longitud_predet_sitio]
- ipv6 spt
- ipv6 spu prefix ifindex [life L]
- ipv6 [-p] gp
- ipv6 [-p] gpu [valor parámetro] ... (try -?)
- ipv6 renew [ifindex]
- ipv6 [-p] ppt
- ipv6 [-p] ppu prefix precedence P srclabel SL [dstlabel DL]
- ipv6 [-p] ppd prefix
- ipv6 [-p] reset
- ipv6 install
- ipv6 uninstall

2 Comandos netsh interface

- 6to4 - Cambia al contexto `netsh interface ipv6 6to4'.
- ? - Muestra una lista de comandos.
- add - Agrega una entrada de configuración en la tabla.
- delete - Elimina una entrada de configuración de una tabla.
- dump - Muestra una secuencia de comandos de configuración.
- help - Muestra una lista de comandos.
- install - Instala IPv6.
- isatap - Cambia al contexto `netsh interface ipv6 isatap'.
- renew - Reinicia las interfaces IPv6.
- reset - Restablece el estado de configuración de IPv6.
- set - Establece la información de configuración.
- show - Muestra información.
- uninstall - Desinstala IPv6.

3 Comandos "netsh interface ipv6 add"

- add 6over4tunnel - Crea una interfaz 6over4.
- add address - Agrega una ruta IPv6 en una interfaz.
- add dns - Agrega una dirección estática del servidor DNS.
- add prefixpolicy - Agrega una entrada de directiva de prefijo.
- add route - Agrega una ruta IPv6 sobre una interfaz.
- add v6v4tunnel - Crea un túnel de punto a punto IPv6-in-IPv4.

4 Comandos netsh interface ipv6 set

- set address - Modifica la información de dirección IPv6.
- set global - Modifca parámetros generales de configuración global.
- set interface - Modifica parámetros de configuración de interfaz.
- set mobility - Modifica parámetros de configuración de movilidad.
- set prefixpolicy - Modifica la información de directiva de prefijo.
- set privacy - Modifica los parámetros de configuración de privacidad.
- set route - Modifica parámetros de ruta.
- set state - Define el estado de la funcionalidad degradada.
- set teredo - Define el estado de Teredo.

5 Comandos netsh interface ipv6 show

- show address - Muestra direcciones IPv6.
- show bindingcacheentries - Muestra entradas de caché de enlace.
- show destinationcache - Muestra las entradas de caché de destino.
- show dns - Muestra las direcciones del servidor DNS.
- show global - Muestra parámetros de configuración global.
- show interface - Muestra parámetros de interfaz.
- show joins - Muestra direcciones de multidifusión IPv6.
- show mobility - Muestra parámetros de configuración de movilidad.
- show neighbors - Muestra entradas en caché de vecinos.
- show prefixpolicy - Muestra entradas de directiva de prefijo.
- show privacy - Muestra parámetros de configuración de privacidad.
- show routes - Muestra entradas de tabla de rutas.
- show siteprefixes - Muestra entradas de la tabla de prefijos de sitios.
- show state - Muestra el estado de la funcionalidad degradada.
- show teredo - Muestra el estado del servicio Teredo.

ANEXO 3 TUNNEL BROKERS

El Tunnel Broker es un mecanismo de transición que habilita a los usuarios a obtener conectividad IPv6 desde cualquier país y lugar, normalmente es gratis, después de registrase en el Tunnel Broker. La comunicación con el Tunnel Broker está basada normalmente en web. Las instrucciones para configurar el sistema del usuario se suelen proporcionar por el propio Tunnel Broker. Este servicio permite crear un túnel IPv6 sobre IPv4 entre un router/host cualquiera y el router del proveedor.

A continuación se muestran algunos de los Tunnel Brokers gratuitos que proporcionan conectividad IPv6.

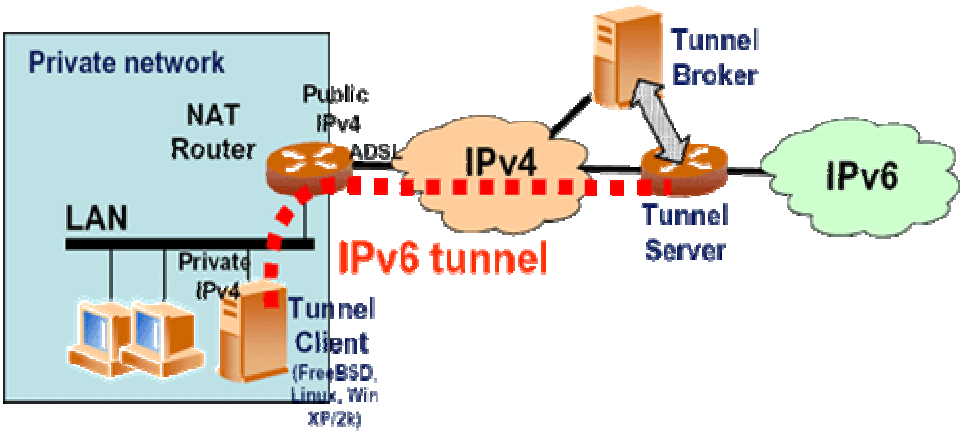
Proveedor del Servicio	URL del servicio
The IPv6 Portal	http://www.ipv6tf.org/using/connectivity/register.php
Access to IPv6 (XS26)	http://www.xs26.net/.
UKERNA	http://www.broker.ipv6.ac.uk/
UK6	http://www.uk6x.com/
AARNet Tunnel Broker	http://broker.aarnet.net.au/

Tabla 13. Proveedores de servicio de Tunnels Brokers

Utilizando direcciones IPv4 privadas como extremo de túneles

Algunos equipos/encaminadores NAT que sólo soportan IPv4, permiten el establecimiento de túneles IPv6 desde sistemas dentro de una LAN privada (empleando direcciones IPv4 privadas) hacia encaminadores o servidores de túneles disponibles en Internet.

El escenario básico de este mecanismo se muestra en la figura siguiente. El cliente del túnel (PC/encaminador) que usa direcciones IPv4 privadas, y que está conectado a Internet a través de un equipo/encaminador NAT sólo-IPv4, puede establecer un túnel IPv6 hacia un servidor de túneles.



El servidor de túneles de 6SOS permite configurar este tipo de conexión si el equipo/encaminador NAT sólo-IPv4 permite el "Reenvío de paquetes Protocolo 41".

Para probarlo:

1. Se solicita un túnel con la dirección IPv4 pública de tu equipo/encaminador NAT.
2. Se configura el cliente de túnel (PC/encaminador), que usa la dirección IPv4 privada, con el script facilitado por nuestro servidor de túneles.

Un solo PC situado detrás del equipo/encaminador NAT, puede obtener conectividad IPv6 con un túnel configurado de esta forma (de un mismo túnel broker). Si se necesita conectividad IPv6 para más de un PC dentro de la red privada, se debe de usar el primer PC (habitualmente el más estable) como cliente del servidor de túneles, y entonces dicho PC se comportará como encaminador para el resto de la red LAN interna IPv6 (se tendrá que haber indicado al túnel broker que el PC es un router). Alternativamente, cada PC debería de usar un túnel broker diferente.

Más información acerca de esta característica: Este mecanismo, conocido como "Reenvío de paquetes Protocolo 41 en equipos NAT" (http://www.ietf.org/internet-drafts/draft-palet-v6ops-proto41-nat-03.txt), permite configurar túneles IPv6 hacia un PC/encaminador interno que utilice direcciones IPv4 privadas. Aunque algunos equipos/encaminadores NAT soportan esta funcionalidad, algunos otros no lo hacen.

En algunos casos se requerirá una configuración adicional para permitir que funcione, mientas que en otros funcionará directamente con la configuración que traen por defecto del fabricante o ISP. Un documento mucho más completo al respecto de este tema está se encuentra en el enlace:
http://www.euro6ix.org/documentation/euro6ix_co_upm-consulintel_wp4_ipv6_tunnels_nat_v1_6.pdf