ESCUELA POLITECNICA DEL EJÉRCITO

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UNA INTRANET IPV6 Y QoS

Previa a la obtención del título de:

INGENIERO DE SISTEMAS E INFORMÁTICA

POR: UBIDIA CARRIÓN ANÍBAL JAVIER

SANGOLQUÍ, 24 de Mayo del 2007

CERTIFICACION

Certifico que el presente trabajo fue realizado en su totalidad por el Sr. ANÍBAL JAVIER UBIDIA CARRIÓN como requerimiento parcial a la obtención del titulo de INGENIERO DE SISTEMAS E INFORMÁTICA.

_____

Fecha

_____

Ing. Fernando Galárraga

DEDICATORIA

A Dios y a mi hija Aylin que siempre me dan alegría y amor

Javier Ubidia

AGRADECIMIENTOS

A Dios por iluminarme y darme fuerzas para seguir adelante. A mis queridos padres por siempre apoyarme incondicionalmente todos estos años. A mi hermano que siempre me contagia de coraje y valor para seguir con mis proyectos. A mi hija que con su sonrisa y mirada tierna siempre me levanta el ánimo en mis peores momentos. A mi esposa y amigos que siempre me han estado incentivándome a culminar mi carrera universitaria.

A la Escuela Politécnica del Ejército, a la Facultad de Sistemas, por permitirme realizarme profesionalmente.

A los directores de tesis, Ing. Fernando Galárraga e Ing. Arturo de la Torre, que sin su apoyo no hubiera sido posible culminar con éxito esta tesis de grado. Y a todos los ingenieros de la Facultad de Sistemas que de una y otra manera me han apoyado incondicionalmente especialmente al Ing. German Ñacato.

Javier Ubidia

INDICE DE CONTENIDOS

INDICE DE TABLAS

INDICE DE FIGURAS

RESUMEN

En esta tesis se ha realizado un análisis de los elementos de software, hardware y comunicaciones que se utilizaron para implementar una Intranet de pruebas con soporte de IPv6 y QoS. Luego de realizado el análisis para ver cual de los elementos de hardware , software y comunicaciones son los mas económicos y los que nos den un mejor desempeño se ha procedido a realizar el diseño de la Intranet de pruebas, la misma que esta compuesta de un servidor en el cual están levantados los servicios de DNS, Web, Correo, FTP y SSH este mismo servidor a su vez actúa como un router central el cual se enlaza a un router esclavo, este router esta conectado a un switch en el cual se han conectado las computadores clientes en donde se realizaron las pruebas respectivas.

En el capitulo IV se detalla de manera grafica todos los pasos que se hicieron para realizar el levantamiento de los diferentes servicios con soporte para IPv6, además se detalla como se realizo la configuración del router y por ultimo se presenta la configuración personalizada del kernel para poder manejar QoS en GNU/Linux. También se presenta los resultados de las pruebas realizadas desde diferentes plataformas a los diferentes servicios levantados en el servidor GNU/LINUX.

CAPITULO I

GENERALIDADES

1.1 Introducción

A principios de los 90, era claro que la Internet iba a ser un proyecto que crecería a pasos inimaginados. Mas y mas direcciones se fueron delegando a un paso alarmante, y para todos estaba muy claro sobre las futuras limitantes que se podrían presentar en cuanto a entidades que pedían conectarse a la siempre creciente red de redes. Pero inicialmente el número no era un problema, IPv4 usa un esquema de direccionamiento de 32bits, por lo tanto el número de host posible es de 2^{32}, lo cual equivale a 4200 millones. El problema real se encuentra en la asignación de direcciones, a pesar de la implementación de estrategias de direccionamiento como CIDR el espacio de direcciones estaba siendo desperdiciado. Adicional a esto, había una necesidad de extender la funcionalidad de la capa de red con características como QoS, encriptación punto a punto, enrutamiento de origen y autenticación entre otros hicieron cada vez mas claro que un nuevo protocolo de Internet tenía que ser adoptado en un futuro cercano.

Tomó muchos años para la IETF (Internet Engeneering Task Force) plantear una nueva versión de IP. Como la IETF produce estándares "abiertos", invitó a toda la comunidad al desarrollo para que dieran a conocer sus necesidades y expectativas de este nuevo protocolo.

Una de las soluciones iniciales al problema de direccionamiento en IPv4 fue conocido como SIPP (Simple IP Plus), donde simplemente se aumentaba el tamaño de las direcciones IP a 64bits y se mejoraban ciertos aspectos de IPv4, como lo eran mejores estrategias de enrutamiento. SIPP era lo mas cercano a lo que la Internet necesitaría después de unas modificaciones. Las direcciones pasaron de 64bits a 128 y se le asignó el nombre de IPv6 (IPv5 ya había sido asignado a otro protocolo, conocido como ST-2, que servia para soporte nativo de ATM en Internet).

1.2 Justificación

El alegato de comenzar a utilizar IPv6 viene dado por que permite dar solución a las necesidades actuales como son la seguridad, movilidad, calidad de servicio (QoS), comunicación de grupo en tiempo real, etc. IPv6 es además extensible y permitirá incorporar nuevos protocolos en el futuro a medida que sean demandados. En la actualidad mediante el manejo de QoS podemos realizar un control de tráfico adecuado dentro de una Intranet. Al implementar QoS en la red podemos solucionar los problemas que traen a la red los programas p2p que están muy de moda y que pueden colapsar el trafico de un red y por otra parte podemos dar mas ancho de banda a ciertos programas que son de importancia crucial o necesitan una latencia baja frente a otros de menor importancia.

Por otra parte la introducción de IPv6 tiene las siguientes ventajas:

1. Aumento del número de direcciones 2^{128}.
2. Mayor velocidad de procesamiento de una cabecera IPv6 básica, no es necesario recalcular el checksun cada vez.
3. Todos los campos en la cabecera IPv6 son de 64 bits, mayores ventajas para la generación actual de procesadores de 64 bits.
4. Los routers no deben examinar las cabeceras de IPv6, salvo la cabecera hop-by-hop.
5. Los routers no realizan fragmentación en IPv6, esto elimina el tiempo de proceso que necesitaban en IPv4 para realizar la fragmentación, con lo cual se obtiene una mayor velocidad de proceso. Sólo los nodos origen pueden realizar la fragmentación.
6. Simplifica la gestión de los ordenadores conectados a la red, soportando de forma automática la conexión de nuevos ordenadores (*plug and play*) sin necesidad de configurarlos explícitamente. La renumeración de redes se simplifica significativamente.
7. Elimina más de la mitad de sus campos de la cabecera del paquete IP para simplificar el diseño de los routers. También ha eliminado las funciones más costosas de procesar, tales como fragmentación, opciones, etc.
8. IPv6 incluye como componente obligatorio el protocolo de seguridad IPsec e integra más eficazmente que IPv4 facilidades tales como distintos grados de calidad de servicio (QoS), movilidad IP, multicast o anycast.

Por todas estas ventajas que presenta IPv6 queda plenamente justificado el que las aplicaciones actuales de Internet estén comenzando a migrar a este nuevo protocolo.

1.3 Objetivos

1.3.1 Objetivo General

Realizar el análisis, diseño e implementación de una Intranet experimental con soporte de IPv6 y QoS utilizando GNU/Linux como sistema operativo para levantar servicios básicos de Internet con el fin de manipularlos para desarrollar pruebas desde clientes con diferentes plataformas.

1.3.2 Objetivos Específicos

- Realizar el análisis de los elementos de hardware, software y equipos de comunicaciones para determinar cuales son los más económicos y los que nos ofrecen mayores y mejores prestaciones.
- Efectuar el diseño de una Intranet de pruebas con soporte de IPv6 y QoS en base a los elementos de hardware, software y comunicaciones que nos arroje el estudio del análisis.
- Hacer la implementación de la Intranet de pruebas con los servicios de Internet y/o servidores de red con soporte de IPv6 y manejo de QoS.
- Efectuar las pruebas a los servicios levantados desde las computadoras clientes para realizar la documentación respectiva.

1.4 Alcance / Meta

Los siguientes puntos describen el alcance/meta de esta tesis:

- Investigar y documentar todo sobre la arquitectura y componentes de IPv6.
- Averiguar y documentar como se realiza QoS en GNU/Linux.
- Configurar el Kernel y utilidades en una distribución GNU/Linux para dar soporte de IPv6 y QoS.
- Especificar como realizar control de tráfico en nuestra red utilizando QoS en GNU/Linux, de manera que podamos gestionar el ancho de banda de nuestra red para cada tipo de tráfico.
- Implementar los principales servicios de Internet y/o servidores de red con soporte para IPv6 en GNU/Linux como son:
  - DNS (servidor de nombres de dominio)
  - DHCP (servidor para asignación de direcciones IPv6 automáticas)
  - FTP (servidor para transferencia de archivos)
  - SMTP (servidor de correos saliente)
  - POP3 (servidor de correos entrante)
  - HTTP (servidor Web)
- Documentar todo el proceso de configuración de los servicios de Internet y/o servidores de red con soporte IPv6 tanto en los clientes como en el servidor.
- Documentar el proceso de configuración del kernel para manejar QoS.
- Realizar pruebas a los servicios levantados y realizar la documentación respectiva.

CAPITULO II

MARCO TEORICO

2.1 El protocolo IPv6

La nueva revisión del protocolo IP se numerará con la versión 6. No se la denominará versión 5 para evitar posibles confusiones, ya que anteriormente a esta revisión se hicieron algunas pruebas añadiendo extensiones a la versión 4. Estas extensiones experimentales no acabaron de formalizarse en una nueva versión del protocolo, con lo que para evitar posibles conflictos se optó por elegir el número 6 para la nueva versión.

La nueva estructura de la cabecera del protocolo IPv6 se caracteriza por tener:

- Direcciones de 128 bits
- Campos de longitud fija

El protocolo IPv6 al igual que IPv4 es un protocolo no fiable y sin conexión, debido a que este sistema funciona y da flexibilidad a la comunicación. Además permite que sean los protocolos de las capas superiores los encargados de mantener un estado de conexión o fiabilidad según sea necesario.

2.1.1 Arquitectura de IPv6

La nueva cabecera del protocolo IPv6 (Figura 2-1) es una evolución de la cabecera IPv4, no se han introducido grandes cambios de estructura, solo se la ha mejorado y optimizado. Se han suprimido algunos campos redundantes u obsoletos y se han ampliado algunas características para hacer frente a las nuevas necesidades de los usuarios como son las comunicaciones en tiempo real y la seguridad.

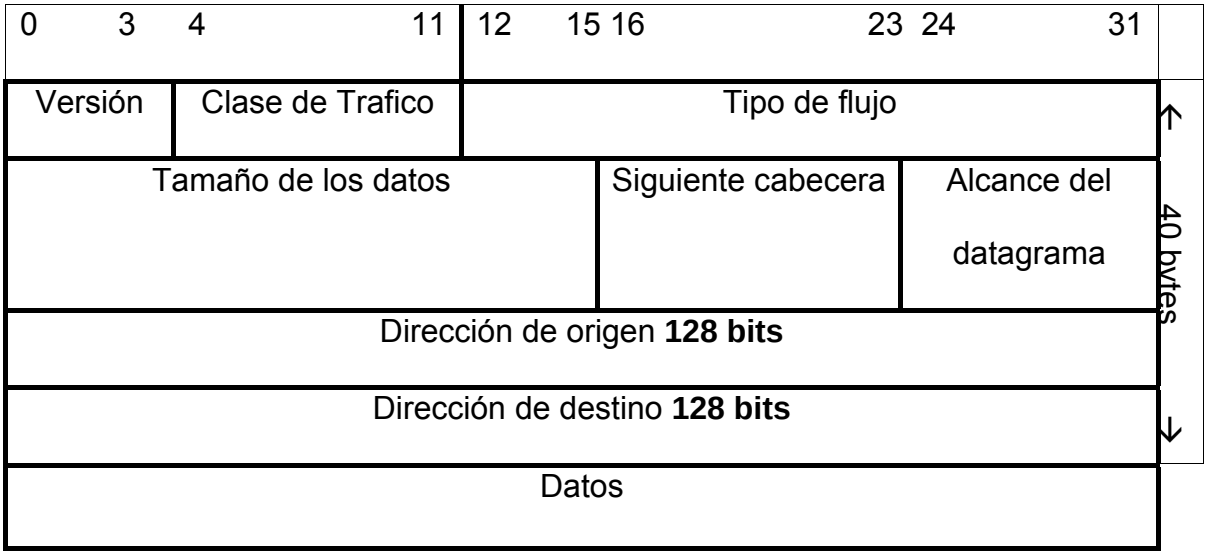


Figura 2-1: Estructura de un datagrama IPv6

Versión (4 bits): Es el primer campo del datagrama. Permite diferenciar que versión de datagrama se recibe (IPv4 o IPv6)

Clase de Trafico (8 bits): Este campo asigna la prioridad del datagrama, una de las nuevas aportaciones para conseguir controlar el flujo de información.

Tipo de flujo (16 bits): Permite especificar que una serie de datagramas deben recibir el mismo trato.

Tamaño de los datos (16 bits): Al igual que en IPv4 especifica el tamaño que tendran los datos, lo que permite un tamaño de máximo 2^{16} = 64K en principio.

Siguiente cabecera (8 bits): Indica al router que tras el datagrama viene algún tipo de extensión. En IPv6 se definen una serie de cabeceras de extensión (Tabla 2-1) que se sitúan fuera del datagrama básico permitiendo al usuario personalizar el tipo de datagrama. Podemos tener varias extensiones de cabecera tan solo indicando en el campo de siguiente cabecera de cada una el tipo de cabecera que vendrá a continuación.

Tabla 2-1: Valores mas usados para el campo siguiente cabecera en IPv6

Valor decimal	Abreviatura	Descripción
0	HBH	Nodo por Nodo
4	IP	IP en IP (encapsulación en IPv4)
5	ST	Stream
43	RH	Cabecera de Encaminamiento (Routing Header)
44	FH	Cabecera de Fragmentación (Fragment Header)
51	AH	Cabecera de Autenticación (Authentication Header)
52	ESP	Encrypted Security Payload
59	NULL	Ninguna cabecera siguiente
60	DO	Destination Options Header
194	JBGR	Jumbogram

Alcance del datagrama (8 bits): Indica el número máximo de routers que puede atravesar un datagrama hasta llegar a su destino. Este campo es el equivalente al tiempo de vida (*TTL*) de la versión 4.

2.1.2 Cabeceras de extensión del protocolo IPv6

En IPv6, la información adicional es codificada en cabeceras que deben ser colocadas en el paquete entre la cabecera IPv6 y la cabecera de la capa de transporte. Las extensiones de cabeceras son identificadas por un valor distinto en el campo siguiente cabecera .Un paquete IPv6 puede contener ninguna, una o varias cabeceras de extensión.

Las cabeceras de extensión tienen una longitud múltiplo de 8 bits, cuando se tiene más de una cabecera de extensión en un mismo paquete, las cabeceras deben aparecer en el siguiente orden:

- Cabecera de Encaminamiento (Routing Header)
- Cabecera de Fragmentación (Fragment Header)
- Cabecera de nodo-por-nodo (Host-by-Host Options Header)
- Cabecera de extremo-a-extremo (End-to-End Options Header)
- Cabecera de Autenticación (Authentication Header)
- Cabecera IPv6 (IPv6 Header)

Cada tipo de cabecera debe aparecer una sola vez en el paquete excepto en el caso de una encapsulación IPv6 en IPv6, donde cada cabecera IPv6 encapsulada debe ser seguida por su propia cabecera de extensión.

2.1.3 Direccionamiento en IPv6

Las direcciones IPv6 son identicadores de interfaces ó conjuntos de interfaces de 128 bits por lo que se tiene tres tipos de direcciones en función del tipo de destino (Figura 2-2):

1. Unicast. Este grupo de direcciones se caracteriza por identificar un único punto final de destino (point-to-point). Un datagrama enviado a una dirección *unicast* será entregado a un solo punto de destino.

2. Multicast. Las direcciones *multicast* agrupan un conjunto de puntos finales de destino. Un datagrama enviado a una dirección *multicast* será entregado a un conjunto de destinos que forman parte de un mismo grupo.

3. Anycast. Este grupo de direcciones al igual que el *multicast* agrupa un conjunto de puntos finales de destino. La diferencia principal con el *multicast* está en sistema de entrega de datagramas. Un datagrama enviado a una dirección *anycast* es entregado solo a un punto de destino (el miembro más cercano del grupo al emisor del datagrama). Este tipo de agrupación no existe en IPv4.

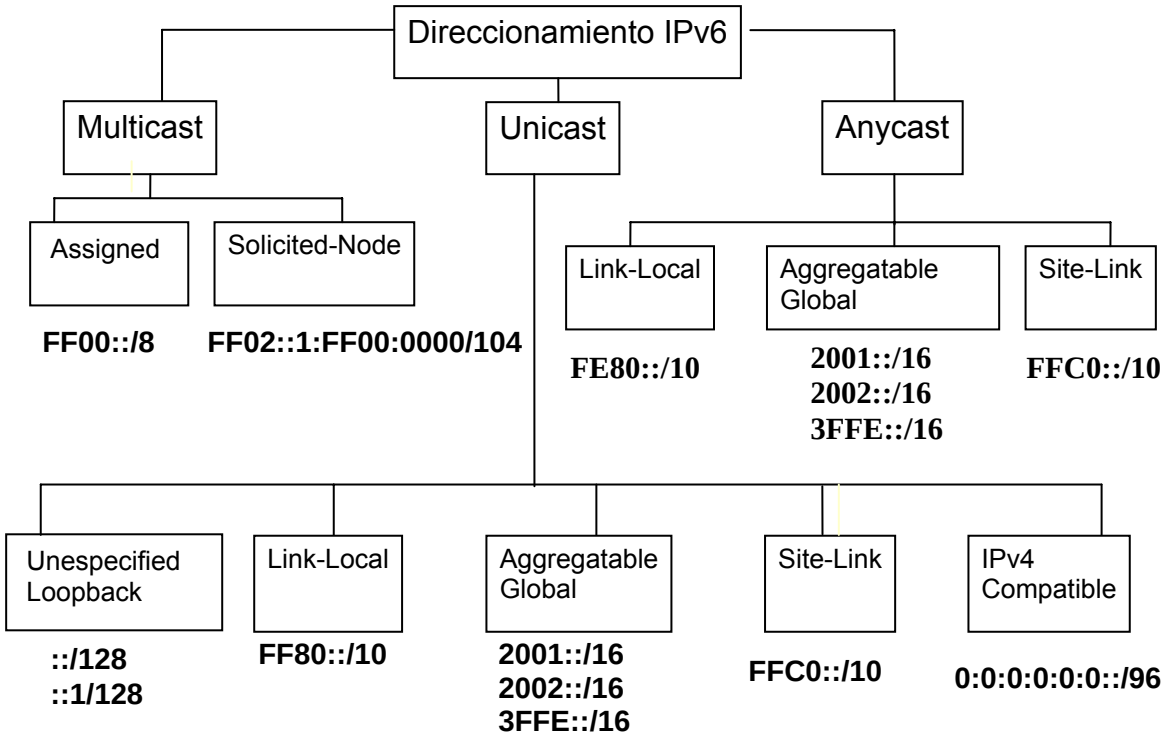


Figura 2-2: Tipos de direcciones IPv6 en función del tipo de destino

También se tienen 3 tipos de direcciones en función del alcance:

- **Link-local**. Solo tienen sentido en el ámbito del enlace.
- **Site-local**. Solo tienen sentido en el ámbito de una organización.
- **Global**. Tienen ámbito global

Figura 2-3: Clasificación de direcciones IPV6 según el alcance

Las direcciones IPv6 están compuestas por 128 bits. La representación de este protocolo viene dada en 8 agrupaciones de 16 bits. De esta forma se puede utilizar la notación hexadecimal, que permite una representación más compacta que tener 128 unos y ceros. La notación general de una dirección IPv6 es:

X:X:X:X:X:X:X:X donde X= 2 octetos en hexadecimal

Ejemplo de dirección IPv6 : ***3ffe:3328:4:3:250:4ff:fe5c:b3f4***

La especificación de un prefijo de direccionamiento en la versión 6 se la realiza mediante la forma *dirección_ipv6/prefijo*.

Ejemplo: ***3ffe:4000:2000::1/64***

Para compactar estas direcciones tan voluminosas, se tienen las siguientes reglas de simplificación:

• Supresión de los ceros redundantes situados a la izquierda mediante el uso del prefijo '**::**'. Este prefijo tan sólo puede ser utilizado una vez en una misma dirección.

Ejemplo: ***FF01:0:0:0:0:0:0:43 → FF01::43***

• Para las direcciones IPv6 obtenidas añadiendo 96 ceros a la dirección IPv4 se permite el uso de notación decimal

Ejemplo: ***10.0.0.1 → 0:0:0:0:0:0:A00:1 → ::10.0.0.1***

2.1.4 Mecanismos de Transición IPv4 – IPv6

Puesto que Internet no va a amanecer un día utilizando IPv6 en vez de IPv4, se han desarrollado una serie de métodos que permitan la convivencia y comunicación entre nodos, sea cual sea su versión de protocolo IP. Entre los mecanismos de transición IPv4 – IPv6 tenemos:

2.1.4.1 Mecanismos Tipo túnel

Encapsulan un paquete IP dentro de otro, es un mecanismo conocido y se usa en la actualidad sobretodo para crear redes privadas virtuales. La utilidad que se le da es para enlazar nubes o islas IPv6 en una Internet basada prácticamente en su totalidad en IPv4.Entre los mecanismos tipo túnel mas utilizados tenemos:

- Túneles estáticos
- Túneles 6to4
- 6over4
- Intra-Site Automatic Tunnel Addressing Protocol(ISATAP)

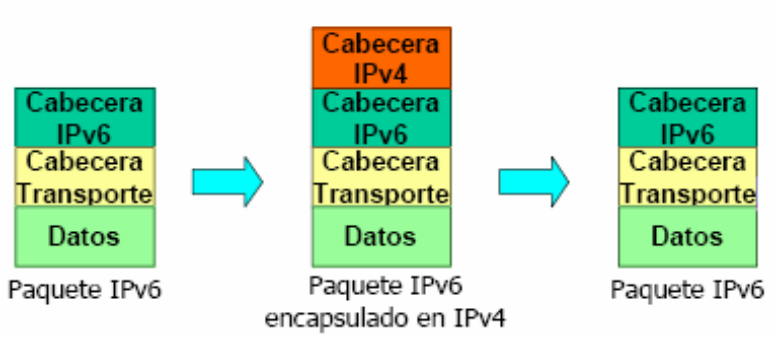


Figura 2-4: Proceso realizado en los mecanismos tipo túnel

2.1.4.2 Mecanismos de Traducción

Su funcionamiento se basa en traducir, en un elemento de red los paquetes de un formato a otro. Algunos de los mecanismos de traducción son:

- Stateless IP/ICMP Translation Algorithm (SIIT)
- Network Address Translation - Protocol Translation(NAT-PT)
- Bump in the Stack (BIS)

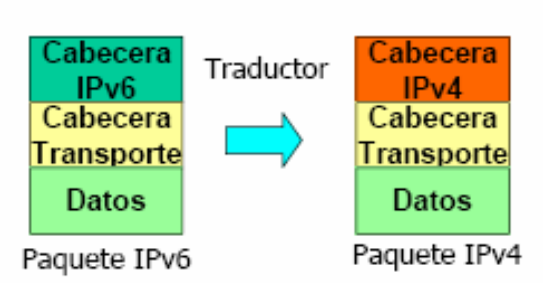


Figura 2-5: Proceso realizado en los mecanismos de traducción

2.1.4.3 SOCKS64

Esta solución es ideal en caso de que se este utilizando ya SOCKS. Con un gateway de tipo SOCKS64 se puede permitir conectar a los clientes tanto a nodos IPv4 como IPv6, sin los típicos problemas asociados a los túneles (fragmentación y límite de saltos).

2.1.4.4 Doble pila

Para que un nodo se pueda comunicar tanto con nodos IPv6 como IPv4, la solución más rápida es pensar en la doble pila de protocolos. Teniendo cada nodo una dirección IPv4 e IPv6 enrutable, se conseguirá que se produzca la comunicación.

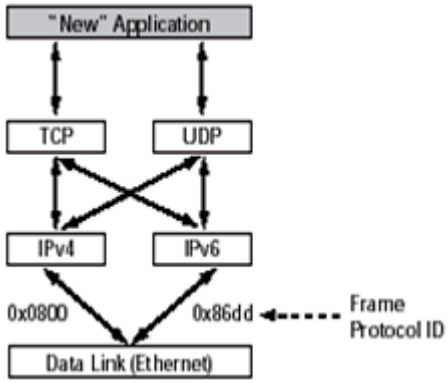


Figura 2-6: Proceso realizado en el mecanismo de doble pila

2.2 El Modelo OSI

El modelo OSI (Open System Interconnection) es un modelo de referencia de 7 capas o niveles. A cada capa se le asigna funciones específicas y las mismas se apilan desde la inferior a la superior de forma que cada una depende de la inmediata inferior para su funcionamiento. Las 7 capas con sus funciones y algunos de los principales protocolos que se utilizan en cada capa se las puede observar en la Tabla 2-2:

Tabla 2-2: Funciones de las capas del modelo OSI

	CAPA	FUNCION	PROTOCOLOS
7	**Aplicación**	Servicios para el usuario como e-mail, servicios de archivos e impresión, emulación de terminal, loguin, etc. Solamente las aplicaciones de PC que trabajen en red encuadra en la capa Aplicación.	DNS, FTP, HTTP, IMAP, IRC, NFS, NTP, POP3, SMTP, SSH, Telnet
6	**Presentación**	Frecuentemente forma parte del sistema operativo y se encarga de dar formato los datos.	XDR, ASN.1, SMB, AFP
5	**Sesión**	Conexión y mantenimiento del enlace	TLS, SSH, RPC, NetBIOS
4	**Transporte**	Realiza el control de extremo a extremo de la comunicación, proporcionando control de flujo y control de errores. Esta capa es asociada frecuentemente con el concepto de confiabilidad.	TCP, UDP, RTP, SCTP, SPX
3	**Red**	Proporciona la posibilidad de rutear la información agrupada en paquetes.	IP, ICMP, IGMP, X.25, ARP, RARP, BGP, OSPF, RIP
2	**Enlace**	Organiza los bits en grupos lógicos denominado tramas o frames. Proporciona además control de flujo y control de errores.	Ethernet, Token Ring, PPP, Frame Relay, ATM, FDDI
1	**Físico**	Define las reglas para transmitir el flujo de bits por el medio físico	cable, radio, fibra óptica

2.3 Servicios de Internet

Servicios de Internet son todas las prestaciones que ofrece el Internet a los usuarios, entre los servicios con que cuenta, tenemos los siguientes:

- **World Wide Web:** Permite consultar información almacenada en cualquier computadora de la red. Es el servicio más flexible, porque además de consultar información permite también enviar datos.
- **SFTP:** Permite el intercambio de archivos de una computadora a otra de forma segura ya que toda la información intercambiada entre su ordenador y el servidor es encriptada.
- **Correo electrónico (e-mail)**: Permite la transferencia personal de mensajes y archivos de un remitente a un destinatario.
- **News:** Son foros de discusión que permiten intercambiar opiniones entre todos los usuarios de Internet.
- **Listas de correo:** Están íntimamente relacionadas con el correo electrónico. Son listas de direcciones electrónicas de personas con intereses comunes.
- **Chat:** Este servicio permite charlar en tiempo real con otros usuarios mediante el teclado de la computadora.
- **Videoconferencia:** Este servicio permite hablar con otra persona de viva voz y viendo además su imagen.
- **SSH:** Al igual que telnet es un servicio de acceso remoto a un servidor de la red con la ventaja de que lo que se transmite a través de esta conexión esta codificado.

2.4 Protocolos de Red

Los protocolos de red son el conjunto de reglas que controlan la secuencia de mensajes que ocurren durante una comunicación entre entidades que forman una red. En este contexto, las entidades de las cuales se habla son programas de computadora o automatismos de otro tipo, tales como dispositivos electrónicos capaces de interactuar en una red. Los protocolos de red establecen aspectos tales como:

- Las secuencias posibles de mensaje que pueden llegar durante el proceso de la comunicación.
- La sintaxis de los mensajes intercambiados.
- Estrategias para corregir los casos de error.
- Estrategias para asegurar la seguridad (autenticación, encriptación).

2.4.1 DHCP

El DHCP (Protocolo de configuración dinámica de servidores) es un protocolo de red en el que un servidor provee los parámetros de configuración a las computadoras conectadas a la Red (máscara, puerta de enlace y otros) y también incluye un mecanismo de asignación de direcciones de IP.

2.4.2 POP3

El Post Office Protocol (POP3) es un protocolo estándar para recibir e-mail.
Se utiliza en clientes locales de correo para obtener los mensajes de correo electrónico almacenados en un servidor remoto.

2.4.3 SMTP

El SMTP o protocolo simple de transferencia de correo electrónico. Es un protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos.

En el conjunto de protocolos TCP/IP, el SMTP va por encima del TCP, usando normalmente el puerto 25 en el servidor para establecer la conexión.

2.4.4 HTTP

El protocolo de transferencia de hipertexto (***HTTP***) es el protocolo usado en cada transacción de la Web (WWW). El hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceder a una página web, y la respuesta de esa web, remitiendo la información que se verá en pantalla. El protocolo http sirve también para enviar información adicional en ambos sentidos, como formularios con mensajes y otros similares.

2.4.5 Protocolos asociados a IPv6

Del conjunto de protocolos de Internet los que están íntimamente ligados a IPv6 son el ICMPv6 y el Descubrimiento del vecindario (ND)

2.4.5.1 Protocolo de Control de Mensajes de Internet (ICMPv6)

El ICMPv6 es un Protocolo de Control de Mensajes de Internet el mismo que es un estándar de IPv6 necesario para que los hosts y los enrutadores que se comunican mediante IPv6 puedan informar de errores y enviar mensajes de eco simples. El ICMPv6, tiene asignado un valor igual a 58 para el campo de "siguiente cabecera".

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

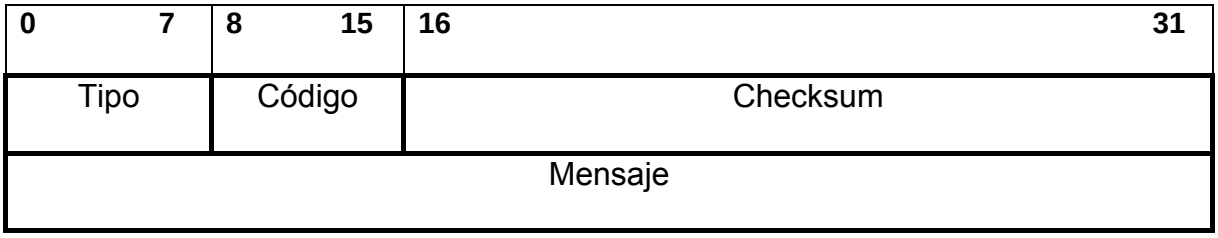


Figura 2-7: Formato genérico de los mensajes ICMPv6

- ***Tipo (8 bits):*** indica el tipo de mensaje, y su valor determina el formato del resto de la cabecera. Los mensajes ICMPv6 se agrupan en dos tipos o clases:

- Mensaje de error: Los mensajes de error tienen cero en el bit de mayor peso del campo “tipo”, por lo que sus valores se sitúan entre 0 y 127.

- Mensajes informativos: Los mensajes informativos tienen valores que oscilan entre 128 y 255.

- ***Código (8 bits):*** depende del tipo de mensaje, y se emplea para crear un nivel adicional de jerarquía para la clasificación del mensaje.

- ***Checksum (32 bits):*** o código de redundancia nos permite detectar errores en el mensaje ICMPv6.

2.4.5.2 Descubrimiento del vecindario (ND)

El protocolo de descubrimiento de vecindario ó Neighbor Discovery (ND) en IPv6 es equivalente al ARP de IPv4. Sin embargo, incorpora también la funcionalidad de otros protocolos IPv4, como “ICMP Router Discovery” y “ICMP Redirect”.

ND es un mecanismo por el cual un nodo que se incorpora a la red, descubre la presencia de otros, en su mismo enlace, para determinar sus direcciones en la capa de enlace, para localizar los routers, y para mantener la información de conectividad acerca de las rutas a los vecinos activos.

El protocolo ND, también se emplea para mantener limpios los “caches” donde se almacena la información relativa al contexto de la red a la que esta conectado un

nodo (host o router), y por tanto para detectar cualquier cambio en la misma. Cuando un router, o una ruta hacia él falla, el host buscará alternativas funcionales.

2.4.6 Protocolos de Routing para IPv6

Se adoptan los mismos protocolos de routing que los existentes en las redes IPv4 pero con algunos cambios indispensables para poder operar con IPv6.A continuación se describirá brevemente los protocolos mas usados en IPv6.

2.4.6.1 RIPng

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

RIPng sólo puede ser implementado en routers. El router incorporará, en la tabla de routing, una entrada para cada destino accesible por el sistema. Cada entrada tendrá como mínimo los siguientes parámetros:

- La métrica o número de saltos (entre 1 y 15).
- El prefijo IPv6 del destino.
- La dirección IPv6 del siguiente router, así como la ruta para llegar a él.

- Un indicador relativo al cambio de ruta.
- Varios contadores asociados con la ruta.

RIPng es un protocolo basado en UDP. Cada router tiene un proceso que envía y recibe datagramas en el puerto 521 (puerto RIPng). El inconveniente de RIPng, al igual que en IPv4, es que:

- Esta orientado a pequeñas redes (máximo 15 saltos) y,
- Su métrica es fija (no varia en función de circunstancias de tiempo real)

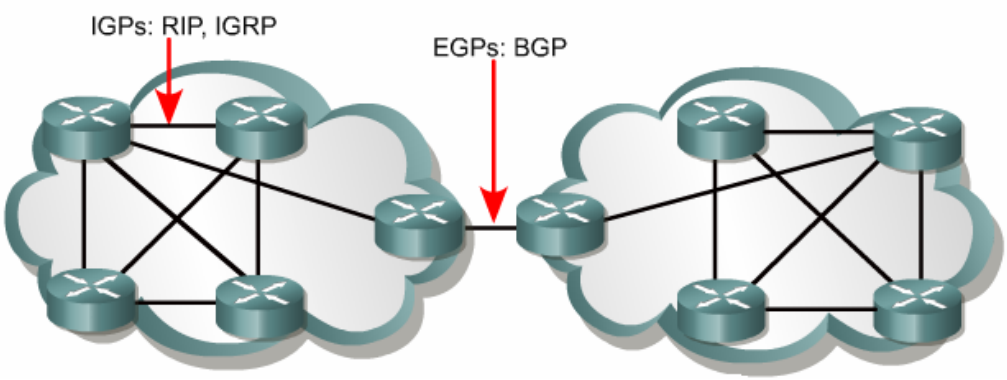


Figura 2.8: Familia de protocolos de routing

2.4.6.2 OSPFv6

El protocolo de rouitng “Abrir Primero el Camino más Corto” (OSPF), es también un protocolo IGP basado en una tecnología de “estado de enlaces”.

Se trata de un protocolo de routing dinámico que detecta rápidamente cambios de la topología (como un fallo en un router o interfaz) y calcula la siguiente ruta

disponible (sin bucles), después de un corto período de convergencia con muy poco tráfico de routing.

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

Los routers distribuyen sus “estados locales” a través de la red por medio de desbordamientos (“flooding”). Todos los routers utilizan el mismo algoritmo en paralelo y construyen un árbol de las rutas más cortas, como si fueran la raíz del sistema. Este árbol de “rutas más cortas” proporciona la ruta a cada destino de la red. Si hubiera varias rutas de igual coste a un determinado destino, el tráfico es distribuido equilibradamente entre todas. El coste de una ruta se describe por una métrica simple, sin dimensión.

OSPFv6 mantiene los mecanismos fundamentales de IPv4, pero se han tenido que modificar ciertos parámetros de la semántica del protocolo, así como el incremento del tamaño de la dirección. OSPFv6 se ejecuta basado en cada enlace, en lugar de en cada subred; además se ha eliminado la autenticación del protocolo OSPFv6, dado que IPv6 incorpora estas características.

2.4.6.3 BGP4+

El Protocolo de Pasarelas de Frontera (BGP – Border Gateway Protocol) es un protocolo de encaminado para la interconexión de redes, es decir, para el routing entre diferentes dominios. Se lo emplea en grandes corporaciones y para la conexión entres proveedores de servicios como ISP.

Su principal función es el intercambio de información de disponibilidad o alcance entre varios sistemas BGP, incluyendo información de la red, permitiendo así construir las rutas más adecuadas y evitar bucles de tráfico.

BGP4+ incorpora mecanismos para soportar routing entre dominios sin clases, es decir, el uso de prefijos, agregación de rutas y todos los mecanismos en los que se basa IPv6.

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

2.5 Servidores de red

Un servidor de red es una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes.

2.5.1 Servidor DNS

El Domain Name System (DNS) es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Aunque como base de datos el DNS es capaz de asociar distintos tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio. ***BIND*** es el servidor DNS más utilizado en Internet.

2.5.2 Servidor WEB

Un servidor Web es un programa que implementa el *protocolo HTTP*. El servidor HTTP ***Apache*** es un servidor HTTP de código abierto multiplataforma, que implementa el protocolo HTTP/1.1, Hosting Virtual (mas de un sitio web en una sola maquina), y soporte de IPv6.

2.6 QoS

QoS (Quality of Service ó Calidad de Servicio) es un conjunto de protocolos y tecnologías que garantizan la entrega de datos a través de la red en un momento dado.

2.6.1 Clasificación de los protocolos de QoS

Las aplicaciones, la topología de la red y la política de QoS dictan qué tipo de QoS es más apropiado para un flujo individual o para varios. De entre todas las opciones, los protocolos y algoritmos más utilizados son:

2.6.1.1 RSVP : Protocolo de reserva de recursos

El Protocolo de Reserva de Recursos es un protocolo de señalización que proporciona un control para la reserva de recursos, orientado fundamentalmente a redes IP.

La reserva de recursos se realiza en los routers intermedios situados a lo largo de toda la ruta de datos de la aplicación. Es la más compleja de todas las tecnologías de QoS para las aplicaciones (hosts) y para los distintos elementos de la red (encaminadores y puentes).

El RSVP utiliza clases de QoS de las cuales las mas utilizadas son:

- Servicios garantizados (Guaranteed Service)
- Servicio de Carga Controlada (Controlled-Load Service)

2.6.1.2 DiffServ : Servicios Diferenciados

Differentiated Services (DiffServ or DS) es un protocolo de QoS que permite dividir y el dar prioridad al tráfico de la red mediante el uso de etiquetas en las cabeceras de los paquetes.

Permite a los proveedores de servicios Internet y a usuarios de grandes redes IP corporativas desplegar rápidamente diferentes niveles QoS en la troncal. A diferencia de RSVP no especifica un sistema de señalización, consiste en un método para marcar o etiquetar paquetes, permitiendo a los routers modificar su comportamiento de envío. Cada tipo de etiqueta representa un determinado tipo de QoS y el tráfico con la misma etiqueta se trata de la misma forma.

Los elementos principales de la arquitectura DiffServ son:

Clasificador: entidad que selecciona paquetes en base al contenido de las cabeceras, según unas reglas definidas.

Medidor: mide el tráfico enviado que se ajusta a un perfil.

Marcador: controla el tráfico mediante el re-marcado de los paquetes con un código diferente (si es necesario).

Modelador: controla el tráfico retardando paquetes para no exceder la velocidad especificada.

Elemento de descarte: descarta paquetes cuando la velocidad de transferencia excede de la especificada.

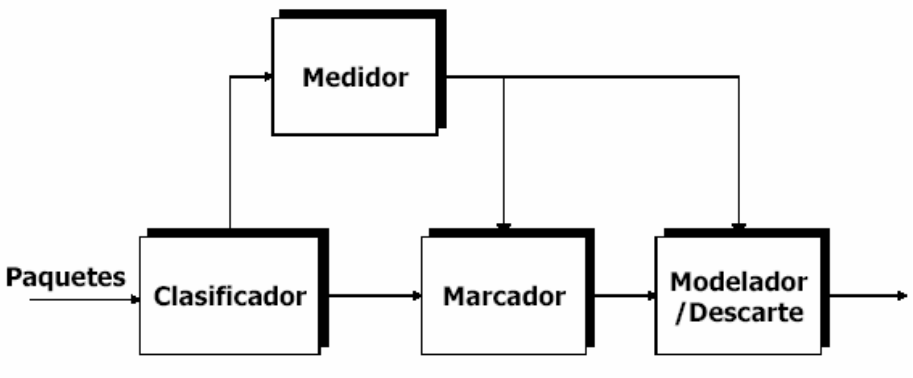


Figura 2-9: Arquitectura de Diffserv

2.6.1.3 MPLS: Conmutación de etiquetas multiprotocolo

MPLS proporciona la posibilidad de administrar el ancho de banda de la red a través de etiquetas en las cabeceras de los paquetes (encapsulamiento) y de encaminadores específicos capaces de reconocerlas.

MPLS es el avance más reciente en la evolución de las tecnologías de routing y forwarding en las redes IP, combina en uno solo la inteligencia del routing con la rapidez del switching.

MPLS usa un esquema de etiquetado del tráfico hacia delante; el tráfico es marcado en su entrada a la red pero no en los puntos de salida. MPLS reside únicamente en los routers y es independiente del protocolo utilizado.

El paquete MPLS (Figura 2-10) tiene una cabecera que contiene 20 bits para etiquetado, un campo de 3 bits para especificar la Clase de Servicio (CoS), 1 bit

que funciona como indicador de etiquetas y un campo de 8 bits que indica el tiempo de vida (TTL).

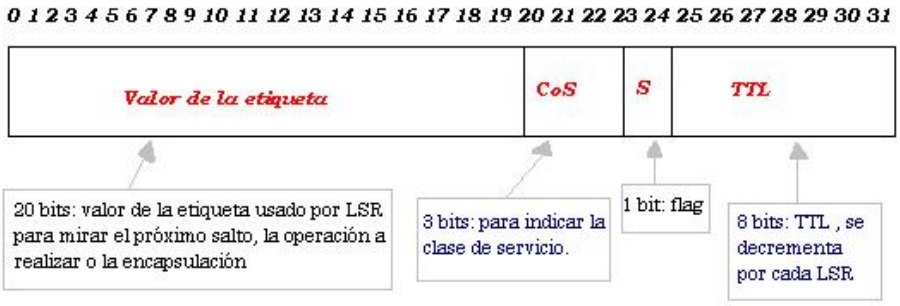


Figura 2-10: Paquete MPLS

2.6.1.4 SBM : Administración del ancho de banda de la subred

Es un protocolo de señalización que permite la comunicación y coordinación entre los distintos nodos de la red, definiendo cómo relacionar los distintos protocolos de QoS superiores con las diferentes tecnologías de capa 2 (capa de enlace en el modelo OSI). Ha sido desarrollado para aplicarlo con LANs IEEE**.**

Existen algunas tecnologías creadas para proporcionar QoS en la capa de enlace, como ATM, pero ésta es una tecnología imposible de implementar por algunas empresas, debido a su coste económico y a su complejidad. Todas estas empresas, por el contrario, utilizan otras tecnologías más comunes para sus LANs, tales como Ethernet , que originalmente no fueron diseñadas para ofrecer QoS.

Ethernet proporciona, simplemente, un servicio análogo al prestado por IP, el servicio Best Effort, en el que existe la posibilidad de que se produzcan retardos y

variaciones (jitter) que pueden afectar a aplicaciones de tiempo real. Por todas estas cosas, IEEE ha redefinido el estándar Ethernet y otras tecnologías de la capa de enlace para proporcionar QoS, mediante diferenciación de tráfico.

Los estándares IEEE 802.1p, 802.1q y 802.1D definen cómo los conmutadores Ethernet pueden clasificar las tramas para poder entregar en primer lugar el tráfico considerado crítico. El grupo de trabajo del IETF para la especificación de las capas de conexión (ISSL) se encarga de definir cómo relacionar los distintos protocolos de QoS de capas superiores con las diferentes tecnologías de la capa 2, como Ethernet. Entre otras cosas, el ISSL ha desarrollado el protocolo SBM para aplicarlo con LANs 802. Un requisito fundamental en SBM es que todo el tráfico debe pasar por lo menos por un conmutador que utilice SBM.

2.7 GNU/Linux

GNU/LINUX conocido como Linux, es un sistema operativo, compatible Unix.
Dos características muy peculiares lo diferencian del resto de los sistemas que podemos encontrar en el mercado:

- Es libre, esto significa que no tenemos que pagar ningún tipo de licencia a ninguna casa desarrolladora de software por el uso del mismo
- El sistema viene acompañado del código fuente.

El sistema lo forman el núcleo del sistema ó kernel más un gran número de programas y librerías que hacen posible su utilización.

Linux se distribuye bajo la Licencia Pública General GNU (GPL), por lo tanto, el código fuente tiene que estar siempre accesible.

2.7.1 Funcionamiento de QoS en Linux

El proceso que sigue un paquete de datos desde que llega hasta una maquina local hasta que abandona la misma es:

● El paquete llega al ***kernel*** linux de la máquina.

● ***Netfilter*** (iptables) se encarga de poner una *marca al paquete* que nosotros establezcamos. Es similar al código postal del sistema de correos: un identificador para posteriormente poder clasificar el envío. Se pueden hacer marcados en base al puerto de destino, a la cabecera IP, la dirección IP origen del paquete.

● El ***árbol de preferencias***: aquí es donde reside el corazón del mecanismo de control de tráfico. Con el encolamiento determinamos de qué forma se envían los datos, mediante el uso de *disciplinas de cola***.**

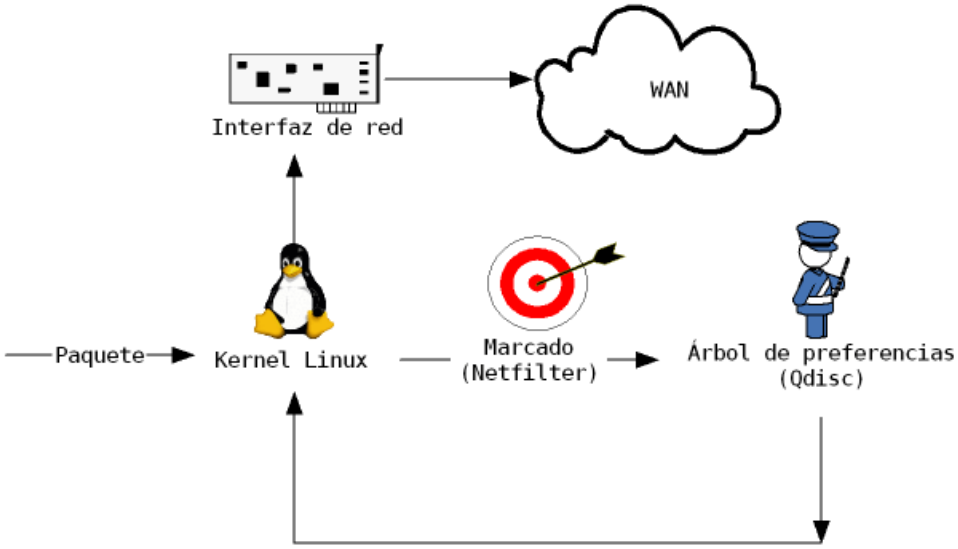


Figura 2-11: Proceso realizado por el kernel de Linux para marcar paquetes

2.7.2 Disciplinas de cola

Las disciplinas de cola son unas estructuras de clases en las que hay dependencia de padre-hijo entre sus miembros. Antes de entrar con detalle en ellas, es necesario conocer los algoritmos que hacen funcionar el mecanismo. A continuación se expone los macanismos más comúnmente usados:

- **ESFQ** (Enhaced Stochastic Fair Queing): el ancho de banda se reparte equitativamente entre todas las conexiones, dando la misma oportunidad a todas para enviar datos. Además, permite hacer el reparto en base a la dirección IP de la conexión. Esto es útil para controlar los programas P2P, que se aprovechan al crear múltiples conexiones.
- **HTB** (Hierarchical Token Bucket): permite dividir el ancho de banda disponible entre un mínimo y un máximo. El algoritmo asegura la disponibilidad del

mínimo, y si se permite, alcanzar el máximo. Además puede “pedir prestado” el ancho de banda sobrante que no se use.

- **SFQ** (Stochastic Fairness Queueing): distribuye el ancho de banda de una determinada interfaz de red de la forma más justa posible dando a cada uno de los flujos de comunicación la oportunidad de enviar sus paquetes por turnos.
- **IMQ** (Intermediate Queing Device): se usa para encolar paquetes y limitar cuánto tráfico puede llegar a la interfaz de red. Tiene utilidad a la hora de limitar el tráfico entrante.
- **WRR** (Weighted Round Robin, Round Robin por Peso): similar al ESFQ, permite dar mas prioridad a una dirección IP concreta y crear jerarquías balanceadas de tráfico.
- **PFIFO_FAST:** First In, First Out: el primero en entrar, es el primero en salir. Es el que está activado por defecto en linux, y el que se aplica también por defecto cuando creamos una clase.
- **RED** (Random Early Detection): utilizado para detectar la congestión. Se asegura de que la cola no se llene.
- **ECN** (Explicit Congestion Notification): funciona en conjunto con RED

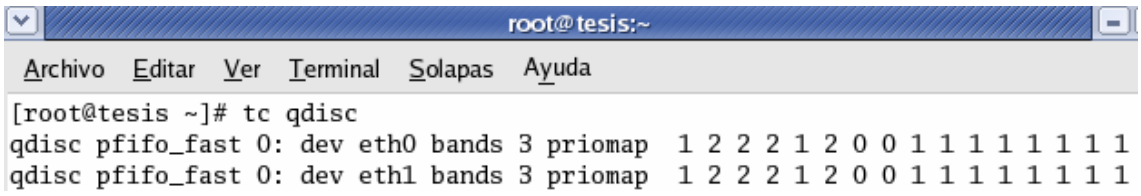


Figura 2-12: Disciplina de colas activada por defecto en linux

2.7.3 Control de Tráfico en Linux

El kernel de Linux a partir de la versión 2.2.X esta diseñado para brindar facilidades como routing, cortafuegos y clasificación de tráfico de manera más potente que algunos productos dedicados como routers y filrewalls.

La herramienta IPROUTE2 es la encargada de permitirnos configurar y gestionar todo este conjunto de nuevas facilidades. IPROUTE2 proporciona básicamente dos herramientas con las que implementan todas estas nuevas facilidades:

- Herramienta *ip*
- Herramienta *tc* (traffic control) : permite implementar la Gestión de Tráfico

El código de control de tráfico en el kernel de Linux consiste en los siguientes componentes principales:

- Disciplinas de cola (queueing disciplines)
- Clases (dentro de una disciplina de cola)
- Filtros (filters)
- Vigilancia (policing y los conceptos relacionados)

2.7.4 Componentes de Control de Tráfico en Linux

Los principales componentes en el control de tráfico en Linux son:

Shaping: Un "shaper" retarda los paquetes para satisfacer una velocidad estipulada. Se utilizan para limitar el tráfico de manera que no supere una velocidad, generalmente para suavizar el tráfico de ráfagas. Utilizan mecanismos de token y bucket.

Scheduling: Un "scheduler" ordena o desordena los paquetes antes de ser desencolados. Scheduling es el mecanismo por el cual los paquetes son ordenados (o desordenados) entre la entrada y la salida de una cola.

Classifying: Los clasificadores (classifiers) ordenan o separan el tráfico entre colas. Clasificación es el mecanismo por el cual los paquetes son separados para tener diferente tratamiento, posiblemente diferente colas de salida.
En Linux el modelo permite que un paquete "fluya" a través de una serie de clasificadores dentro de una estructura de control de tráfico y que sea clasificado de acuerdo a las políticas (policing).

Policing: Policing, como un elemento del control de tráfico, es simplemente un mecanismo por el cual el tráfico es medido y limitado. Un policer es una pregunta tipo si/no acerca de que hacer con el tráfico que ingresa en una cola. Si bien un policer utiliza mecanismos de token bucket no tiene la capacidad de retardar el tráfico como un "shaper".

Dropping: Descartar un paquete, un flujo o una clasificación.

Marking: Es el mecanismo por el cual un paquete es alterado o marcado. Los mecanismos de marcado del control de tráfico instalan una marca en el paquete mismo, la cual es usada y respetada por otros routers dentro de un mismo dominio administrativo (DiffServ).

Tabla 2-3: Componentes del Control de Tráfico en Linux

Elemento Tradicional	**Componente de Linux**
Shaping	Una *class* ofrece capacidades de shaping.
qdisc	Una *qdisc (queue discipline)* es un scheduler.
Classifying	El objeto *filter* realiza la clasificación utilizando un objeto *classifier*. En Linux los *classifiers* no pueden existir fuera de los *filter.*
policing	Un *policer* solo existe como parte de un *filter*.
dropping	Para hacer drop del tráfico se requiere un *filter* con un *policer* que utilice "drop" como acción.
marking	Se utiliza *dsmarkqdisc*.

2.7.5 Disciplinas de cola en el control de tráfico

Las disciplinas de cola siguen una estructura jerárquica en árbol. Cada interfaz (tarjeta de red) tiene una “qdisc raíz” que esta asociada a ella. Cuando se reciben paquetes o tramas de datos en la interfaz de red, la qdisc raíz recibe la petición de desencolar. En base a las marcas en el paquete que hayamos establecido con netfilter / iptables, la qdisc raíz lo enviará a alguna de las clases que contiene. A cada qdisc y cada clase se les asigna un controlador que consiste en dos partes separadas por dos puntos, un número mayor y un número menor. Las clases deben tener el mismo número mayor que sus padres; el número menor debe ser único dentro de una qdisc y sus clases.

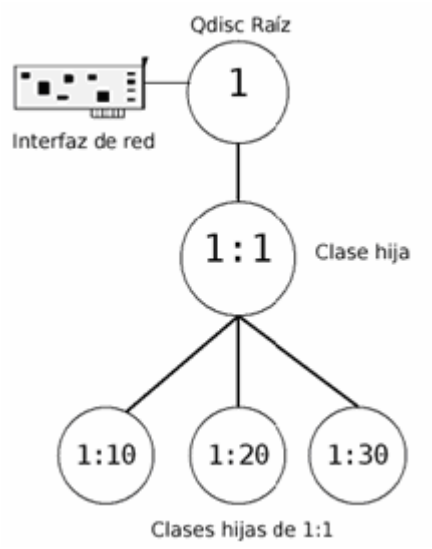


Figura 2-13: Ejemplo de una jerarquía en árbol

CAPITULO III

ANÁLISIS Y DISEÑO DE UNA INTRANET IPV6 Y QOS

3.1 Análisis de una intranet con soporte IPv6 y manejo de QoS

La implementación de este proyecto en una red real seria demasiado riesgosa por cuanto se tienen muchos sistemas en producción es por esto que se ha optado por realizar un análisis para posteriormente diseñar e implementar una red de pruebas en donde mediante pruebas de rendimiento veremos las ventajas y desventajas de utilizar una red con IPv6.

Para realizar el análisis de los elementos que se utilizaran en la intranet de pruebas presentaremos un pre-diseño (Figura. 3-1) en el cual se utilizaran los siguientes elementos activos:

- 1 Servidor
- 1 Router
- 1 Switch
- Varios PC clientes

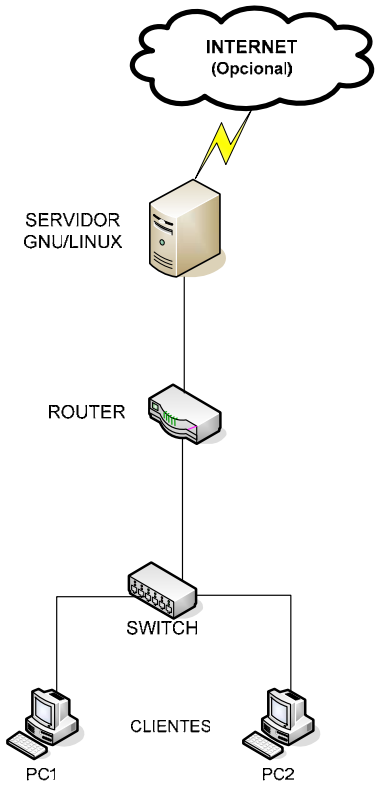


Figura 3-1: Pre-diseño de la red de pruebas

El análisis realizado consiste en determinar los elementos de hardware, software y comunicaciones que se utilizarán al implementar la intranet de pruebas con soporte de IPv6 y manejo de QoS, con este propósito se realizaran tablas comparativas para poder evaluar cual es la mejor opción.

3.1.1 Análisis de Software

3.1.1.1 Sistema Operativo

Para el análisis del sistema operativo a utilizar se han tomado dos opciones la una basada en software libre y la otra basada en software comercial ambas tomando en cuenta las versiones mas actuales y estables.

Las opciones en sistemas operativos comerciales que se han elegido son Windows 2003 Server, Windows XP, Windows Vista y como sistemas operativos de distribución libre a Centos 4.3 y Fedora Core 5. Se ha elegido Centos 4.3 por ser un clon a nivel binario de la distribución Red Hat Enterprise Linux este lo compararemos con Windows 2003 Server puesto que ambos son orientados a instalaciones en servidores.

Para análisis del sistema operativo que se utilizara tanto en el servidor como en los clientes tomaremos en cuenta características como si posee soporte nativo de IPv6 y manejo de QoS frente al costo de las licencias de cada uno de estos sistemas operativos.

Tabla 3-1: Comparación de sistemas operativos

Sistema operativo	**Usado en aplicaciones**	**Tipo de distribución**	**Soporta IPv6 nativo**	**Soporta QoS nativo**	**Disponible en Internet**	**Costo de licencia**
Fedora Core 5	Servidor ó Cliente	Libre	Si	Si	Si	Gratis
Centos 4.3	Servidor ó Cliente	Libre	Si	Si	Si	Gratis
Windows XP SP2	Cliente	Comercial	Si	Si	No	$110
Windows Vista R1C	Cliente	Comercial	Si	Si	Si (versión Beta)	$150
Windows 2003 Server	Servidor	Comercial	Si	Si	No	$400 (5 usuarios)

Del análisis de la Tabla 3-1 se desprende que la mejor opción de sistema operativo es Centos 4.3 para servidores y Fedora Core 5 para los clientes por cuanto tienen soporte de IPv6 y QoS nativos y el costo de la licencia es nulo pudiéndoselo bajar de Internet libremente.

Luego de realizar el análisis de soporte IPv6 y QoS en los diferentes sistemas operativos analizaremos que mecanismos de QoS soporta cada uno, si son administrables por el usuario y si pueden manejar QoS con IPv6.

Tabla 3-2: Análisis de soporte IPv6/QoS en los sistemas operativos

Sistema operativo	**Mecanismos de QoS soportados**	**Los mecanismos de QoS son administrable por el usuario**	**Soporta Qos en IPv6**
Fedora Core 5	RSVP RSVP para IPv6 Packet Classifer Packet Scheduler QoS and/or fair queueing TC Traffic policing Diffserv field marker U32 classifier	Si	Si
Centos 4.3	RSVP RSVP para IPv6 Packet Classifer Packet Scheduler QoS and/or fair queueing TC Traffic policing Diffserv field marker U32 classifier	Si	Si
Windows XP con SP2	RSVP Packet Classifer Packet Scheduler GQoS	No	No
Windows 2003 Server	RSVP Packet Classifer Packet Scheduler GQoS	No	No

Analizando la Tabla 3-2 podemos decir que la mejor opción sigue siendo Fedora y Centos como sistemas operativos ya sean aplicados a servidores o clientes, puesto que manejan mecanismos de QoS administrables por el usuario y tienen soporte para IPv6.

3.1.1.2 Clientes con soporte IPv6

Para la prueba de los servidores y servicios de Internet debemos utilizar algunos programas que soporten los protocolos correspondientes a estos servicios, para el análisis de los mejores programas haremos una tabla comparativa en la cual analizaremos principalmente si tienen soporte para IPv6 y si son de distribución libre, además de las plataformas que soportan.

Tabla 3-3: Clientes con soporte de IPv6

Cliente	Paquete / Versión	Soporta IPv6	Plataforma soportada
Web y FTP	Mozilla Firefox / 1.5.0.4	Si	Windows y Linux
	Internet Explorer / 7.0	Si	Windows
Correo	Thunderbird / 1.5.0.9	Si	Windows y Linux
	Sylpheed / 2.2.5	Si	Windows y Linux
	Windows Mail / 6.0	Si	Windows
	Evolution / 2.0.2	Si	Linux
SSH	PuTTY / 0.59	Si	Windows
DHCP	Dibbler / 0.4.1	Si	Windows y Linux

3.1.1.3 Analizadores de protocolos y tráfico de red

Entre los analizadores de protocolos y tráfico de red que utilizaremos tendremos a Ethereal (actualmente llamado Wireshark) como analizador de protocolos y tcpdump como analizador de tráfico de red, a continuación se presenta las principales características de estos programas utilitarios.

Tabla 3-4: Herramientas de análisis de tráfico

Software	Tipo	Tipo de Interface	Plataforma soportada	Características principales del programa
Ethereal 0.99 ó Wireshark 0.99.3	Libre	Grafica	Windows / Linux	Es un analizador de protocolos, utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos
Tcpdump 3.9.4	Libre	Línea de comandos	Windows / Linux	Es una herramienta en línea de comandos cuya utilidad principal es analizar el tráfico que circula por la red.

3.1.1.4 Generadores y consumidores de tráfico

En cuanto a este tipo de software analizaremos algunas opciones basándonos principalmente en que si pueden generar trafico IPv6 y el tipo de distribución y plataformas que este software soporta.

Tabla 3-5: Programas de generación y consumo de tráfico de red

Software	Tipo de distribución y plataformas soportadas	Tipo de trafico generado/ consumido	Características principales del programa
Mgen	Libre solo linux	UDP (tanto para IPv4 como para IPv6)	Genera patrones de tráfico en tiempo real para destinos unicast/multicast según un script de configuración o por línea de comandos. Se pueden especificar tamaño de paquetes y velocidades de transmisión para flujos individuales.
Netperf	Libre solo en linux	TCP (tanto para IPv4 como para IPv6)	Netperf es una prueba patrón que se utiliza para medir el funcionamiento de diversos tipos de redes
Iperf	Libre funciona en Windows y linux	TCP/UDP (tanto para IPv4 como para IPv6)	Es una herramienta que sirve para medir el desempeño de una red.
IPTraf	Libre solo linux	TCP/UDP (solo IPv4)	Es un monitor de consumo de ancho de banda
Ethloop	Libre solo linux	TCP/UDP (solo IPv4)	Es un generador de paquetes

Revisando la Tabla 3-5 se concluye que los mejores candidatos en cuanto a software de generación y consumo de trafico son mgen, netperf e Iperf por generar trafico tanto IPv4 e IPv6 y ser de distribución libre.

3.1.1.5 Herramientas de administración, configuración y acceso remoto

Entre las herramientas de configuración que utilizaremos tendremos a Webmin como herramienta de configuración remota y VNC como herramienta de acceso remoto puesto que nos ayudaran a realizar la configuración de los equipos de forma rápida y utilizando un solo Terminal de control. A continuación se presenta un cuadro con las principales características de estas herramientas estas características seguirán basándose en lo que hemos venido analizando en todo software a utilizar que es si es de distribución libre.

Tabla 3-6: Herramientas de configuración y acceso remoto

Software	Tipo de uso	Tipo de distribución	Características principales del programa
Webmin	Administración y configuración	Libre	Webmin es un interface web para administrar y configurar un sistema Unix usando cualquier navegador que soporte tablas y formularios. Consta de un simple servidor web y un sinfín de scripts Perl5 y no usa módulos externos.
VNC	Acceso remoto	Libre	Software cliente/servidor que permite acceder remotamente a sesiones X−Windows. Con este programa se puede acceder desde cualquier ordenador conectado a Internet que tenga el cliente (vncviewer) a una sesión que abierta en el ordenador.

3.1.1.6 Software de ruteo

Tabla 3-7: Características principales de zebra y quagga

Software	Características principales
Zebra	Zebra es un software para plataformas linux el cual se encarga de manejar las tablas de ruteo en estos sistemas, realizando las funcionalidades de un router físico. Es excelente pues maneja los protocolos de ruteo como RIPv1, RIPv2, RIPng, OSPFv2, OSPFv3, BGP-4 y BGP-4+ y tiene soporte para IPv4 e IPv6.
Quagga	Es un paquete de software de encaminamiento avanzado derivado de zebra que proporciona los protocolos de encaminamiento basados en TCP/IP. Maneja protocolos de ruteo como RIPv1, RIPv2, RIPng, OSPFv2, OSPFv3, BGP-4 y BGP-4+ tanto para IPv4 e IPv6.

Considerando que las características que se muestra en la Tabla 3-7 se puede ver que son iguales, por tanto se puede utilizar cualquiera de los dos paquetes de software antes mencionados pues ambos son de distribución libre y funcionan bajo plataformas linux y nos permite manejar protocolos de ruteo tanto en IPv4 como en IPv6.

3.1.2 Análisis de Hardware

En cuanto al hardware a ser utilizado tendremos que tomar en cuenta los requerimientos mínimos y recomendados que cada sistema operativo requiere para operar con normalidad y evaluaremos cual de ellos requiere menores requerimientos de hardware.

Tabla 3-8: Requerimientos para la instalación de Centos 4 y Fedora Core 5

Centos 4.3 y Fedora Core 5	
Requerimientos mínimos	**Requerimientos recomendados**
Procesador: Intel Pentium I/II/III/IV/Celeron/Xeon, AMD K6/II/III, AMD Duron, AMD Athlon/XP/MP, Advanced Micro Devices AMD64(Athlon 64, etc) e Intel EM64T (64 bit), PPC ó Procesadores Alpha	
Memoria RAM: 64 MB	Memoria RAM: 128MB
Espacio en Disco Duro: 512 MB	Espacio en disco duro : 2 GB
Unidad de CD-ROM o DVD-ROM (requerida para instalaciones con CD)	Unidad de CD-ROM o DVD-ROM 12x o más rápida

Tabla 3-9 Requerimientos para la instalación de Windows 2003 Server

Windows 2003 Server	
Requerimientos mínimos	**Requerimientos recomendados**
Procesador Intel Pentium (o compatible) 133 MHz	Procesador Intel Pentium II (o compatible) a 550Mhz o superior
128 MB en RAM	256MB en RAM
Espacio disponible en disco de 1.5 GB para la instalación	2GbB de espacio disponible en disco
Unidad de CD-ROM o DVD-ROM (requerida para instalaciones con CD)	Unidad de CD-ROM o DVD-ROM 12x o más rápida

Tabla 3-10: Requerimientos para la instalación de Windows XP

Windows XP	
Requerimientos mínimos	**Requerimientos recomendados**
Procesador Intel Pentium (o compatible) a 233 MHz o superior	Procesador Intel Pentium II (o compatible) a 300 MHz o superior
64 MB en RAM	128 MB en RAM (máx. 4GB en RAM)
Disco duro de 2 GB con espacio disponible en disco de 650 MB	2 GB de espacio disponible en el disco duro
Unidad de CD-ROM o DVD-ROM (requerida para instalaciones con CD)	Unidad de CD-ROM o DVD-ROM 12x o más rápida

Observando las tablas de requerimientos anteriores vemos que Centos y Fedora nos dan una amplia posibilidad de arquitecturas de procesadores y soportan todas las velocidades de procesadores en la arquitectura Intel con un mínimo de espacio en disco duro y memoria RAM.

3.1.3 Análisis de equipos de Comunicaciones

3.1.3.1 Router

Para el análisis del equipo de enrutamiento se ha realizado la comparación entre equipos comerciales de las marcas Cisco y 3Com que son las más reconocidas a nivel mundial en cuanto a equipos de comunicaciones frente a un computador de escritorio con sistema operativo Centos y zebra como software de ruteo. Las comparaciones que haremos serán básicamente el soporte que tienen para manejo de IPv6 y QoS frente a los costos que estos equipos presentan.

Tabla 3-11: Análisis de Routers con soporte de IPv6 y QoS

Protocolos de ruteo soportados	Manejo de QoS y control de trafico en IPv6	Equipo	Modelo	Versión de sistema operativo instalado
RIPv1, RIPv2, OSPFv2,OSPFv3, BGP-4, RIPng, BGP-4+	Si	CISCO	Cisco 7500 Cisco 7200 Cisco 7100 Cisco 6400 Cisco 3600 Cisco 2600 Cisco 2500 Cisco 1700	IOS 12.4 (para soportar IPv6)
RIPv1, RIPv2, OSPFv2,OSPFv3, BGP-4	No	3COM	Router 6000 Router 5000 Router 3000	
RIPv1, RIPv2, RIPng, OSPFv2, OSPFv3, BGP-4 y BGP-4+	Si junto con las herramientas de control de trafico del S.O Centos	PC de escritorio	Depende del sistema operativo	Zebra 0.94

Tabla 3-12: Comparación de costos de routers

Equipo	Costo unitario
PC Servidor	$ 400
PC Ruteadores	$ 300
Router 3COM	$ 1100 – 2200
Router CISCO	$ 1800 - 2800

Dado el costo que presenta un router comercial (Tabla 3-12) con manejo de QoS se ha optado por una solución mucho mas barata y flexible la cual consiste en hacer ruteadores mediante computadores de escritorio con sistema operativo Linux y Zebra como software de ruteo, este software posee comandos de

configuración similares a los utilizados en la configuración de equipos CISCO y maneja protocolos de ruteo tanto en IPv4 e IPv6.

El beneficio de utilizar ruteadores mediante maquinas Linux es a parte del menor costo el que se tiene mayores prestaciones puesto que se puede manipular código y hacerlo funcionar como mejor nos de resultado.

3.1.3.2 Switch

Al igual que en los router se ha escogido tres marcas comerciales, para realizar la comparación de los switch nos basaremos en características como si son administrables y permiten realizar VLANs frente al costo de estos equipos.

Tabla 3-13: Características relevantes y costos de switch

Marca	Modelo	Costo	Características relevantes de los switch
CNET	CSH-800W Smart Access	$56	- 8 Puertos 10/100 , todos con detección automática y auto MDI/MDX - Switching de Capa 2, capacidad de switching de 1,5 Gbps - Soporte de VLAN: 8 grupos - Control de tráfico: Control de flujo full-duplex IEEE 802.3X - Administración: Interfaz Web

CISCO	Familia ASA 5500	$110	- 8 Puertos 10/100, con detección automática y auto MDI/MDX - Rendimiento: Switching de Capa 2, - Soporte de VLAN: 25 VLANs (IEEE 802.1Q) - Control de tráfico: Control de flujo full-duplex IEEE 802.3X, control de flujo de la presión trasera para el modo half-duplex, supresión de broadcast storms (umbral de 3.000 pps), soporte de 128 grupos de filtros multicast - Asignación de colas de prioridad: Cuatro colas hardware por puerto, asignación de colas WRR (round robin ponderada) - Priorización de tráfico: DiffServ , prioridad de puertos IP, limitación de velocidades. - Administración: Línea de comando.
3COM	Switch 3COM OC9P	$88	- 8 Puertos 10/100, un puerto de uplink 10/100/1000 Ethernet, todos con detección automática y auto MDI/MDX; un puerto de consola RS232 - Rendimiento: Switching de Capa 2, capacidad de switching de 3,6 Gbps - Soporte de VLAN: 60 VLANs (IEEE 802.1Q) - Control de tráfico: Control de flujo full-duplex IEEE 802.3X, control de flujo de la presión trasera para el modo half-duplex, supresión de broadcast storms (umbral de 3.000 pps), soporte de 128 grupos de filtros multicast - Asignación de colas de prioridad: Cuatro colas hardware por puerto, asignación de colas WRR (round robin ponderada) - Priorización de tráfico: Prioridad por defecto, DSCP (DiffServ CodePoint), prioridad de puertos IP, limitación de velocidades, priorización de voz NBX - Administración: Interfaz Web, de línea de comando, Telnet; administración SNMP v1-3 remota

3.2 Diseño de la intranet con soporte IPv6 y manejo de QoS

Después de realizar el análisis de anterior nuestra red de pruebas constara de los siguientes elementos (Figura. 3-2):

- 1 Servidor
- 1 Router
- 1 Switch
- Varios PC clientes

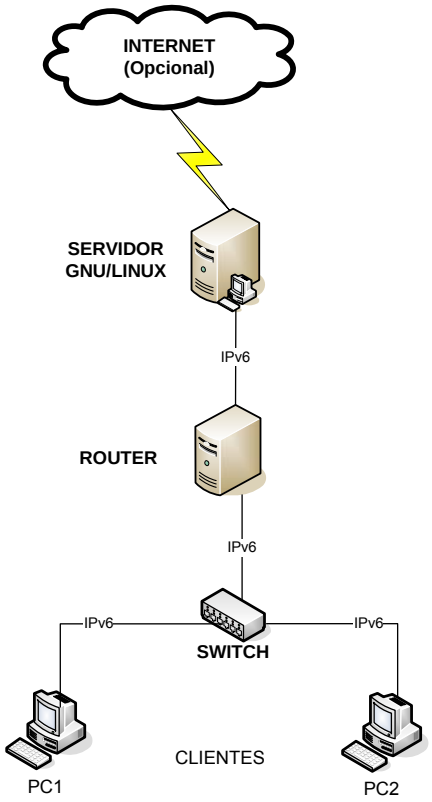


Figura 3-2: Diseño final en base a routers en Linux

El servidor donde se levantaran los diferentes servicios de Internet con soporte de IPv6 y manejo de QoS tendrá como sistema operativo Centos 4.3.

El router será implementado en un computador de escritorio con sistema operativo Centos 4.3 y Zebra como software de ruteo, este computador deberá tener como mínimo 2 tarjetas de red puesto que actuaran como router.

Los clientes que se conectaran a un switch de marca CNET ó 3Com, estos PC's clientes tendrán como sistema operativo Fedora Core 5 ó Windows XP.

3.2.1 Asignación de direcciones IP

A continuación se presenta un cuadro en el cual se presenta como se asignaran las direcciones IP tanto en el servidor, router y en las computadores clientes, las direcciones IP en los PC's clientes serán asignadas mediante el servidor de DHCP dentro del rango presentado en la Tabla 3-14, en cambio el router y el servidor tendrán IP fijas las cuales serán configuradas manualmente en cada equipo.

Tabla 3-14: Rango de direcciones IPs a ser asignadas a los equipos

Hardware	IPv6/prefijo
PC Servidor PC Router	3ffe:4000:2000::1 /64 3ffe:4000:2000::2 / 64
PC´s clientes	3ffe:100:100::2– 50 /64

3.2.2 Características técnicas de los equipos a utilizar

Las características técnicas de los equipos que se poseen para implementar el diseño de la red de pruebas están dentro del rango de los requerimientos analizados anteriormente para que trabajen con normalidad por lo cual nos sirven plenamente para realizar esta tesis. A continuación se detalla cada uno de los elementos que intervendrán en la implementación de la intranet de pruebas y sus respectivas características técnicas:

Tabla 3-15: Características técnicas de router central

Servidor de servicios de Internet
Procesador: Intel Pentium IV de 3GHz
Memoria RAM: 1 GB
Disco Duro: 40GB
Unidad de DVD-ROM 16x
Sistema operativo: Centos versión 4.4
Software de ruteo: Zebra versión 0.94
3 tarjeta de red 10/100 Mbps

Tabla 3-16: Características técnicas de PC's clientes

Computadoras Clientes
Procesador: Intel Pentium IV de 2GHz
Memoria RAM: 512 MB
Disco Duro: 40GB
Unidad de DVD-ROM 16x
Sistema operativo: Windows XP y Fedora Core 5
1 tarjeta de red 10/100 Mbps

Tabla 3-17: Características técnicas de routers esclavos

Router
Procesador: Intel Pentium III de 1GHz
Memoria RAM: 256 MB
Disco Duro: 10GB
Unidad de DVD-ROM 16x
Sistema operativo: Centos versión 4.4
Software de ruteo: Zebra versión 0.94
2 tarjetas de red 10/100 Mbps

Tabla 3-18: Características técnicas de switch administrable

Switch
- Marca : CNET - 8 Puertos 10/100 , todos con detección automática y auto MDI/MDX - Switching de Capa 2, capacidad de switching de 1,5 Gbps - Soporte de VLAN: 8 grupos - Control de tráfico: Control de flujo full-duplex IEEE 802.3X - Administración: Interfaz Web

CAPITULO IV

IMPLEMENTACIÓN Y PRUEBAS DE LA INTRANET IPV6 Y QoS

4.1 Bitácora de instalación del software

A continuación se describe todos los detalles de instalación del software que se utilizo para la implementación de este proyecto.

4.1.1 Instalación de sistema operativo

Para la implementación de la Intranet de pruebas se debe primero instalar el sistema operativo tanto en el servidor, router y clientes. El sistema operativo que se ha instalado en el servidor y router es CentOS 4.3, en los clientes se instalo Fedora Core 5 y Windows XP SP2. Los pasos para realizar la instalación del sistema operativo CentOS en las PC´s son:

1) Bajar una imagen del sistema operativo de Internet (www.centos.org), grabarla en un DVD, bootear el sistema con el DVD y comenzar la instalación.

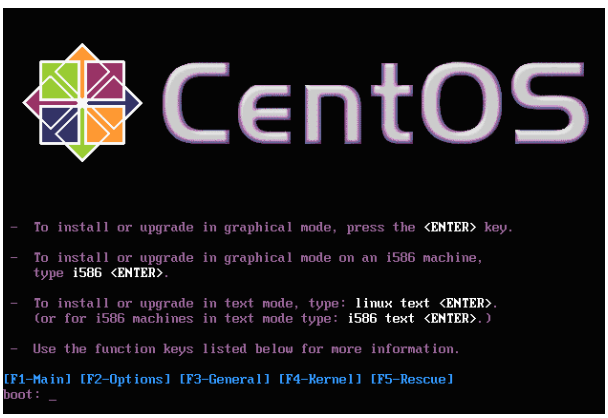


Figura 4-1: Pantalla inicial al bootear con el DVD de instalación de CentOS

2) En la pantalla de bienvenida de CentOS damos click en el botón *Next*.

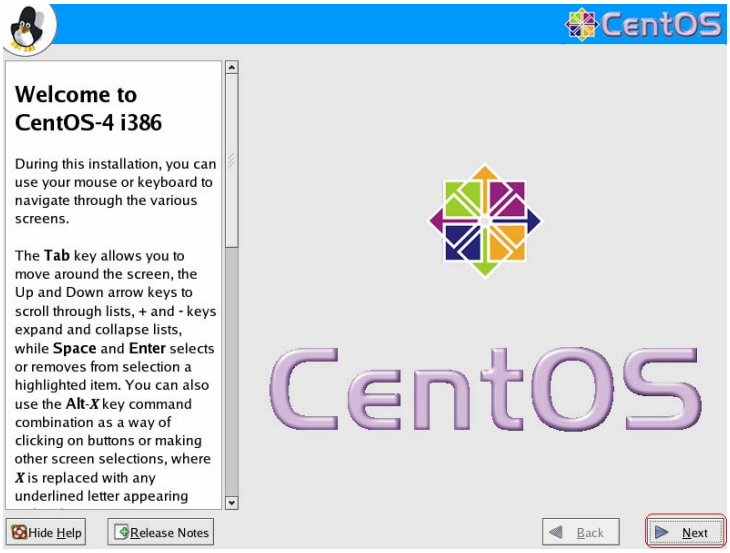


Figura 4-2: Pantalla de bienvenida de CentOS

3) Seleccionamos el lenguaje de instalación en este caso se ha escogido *Spanish(Español)* y damos click en el botón *Next*.

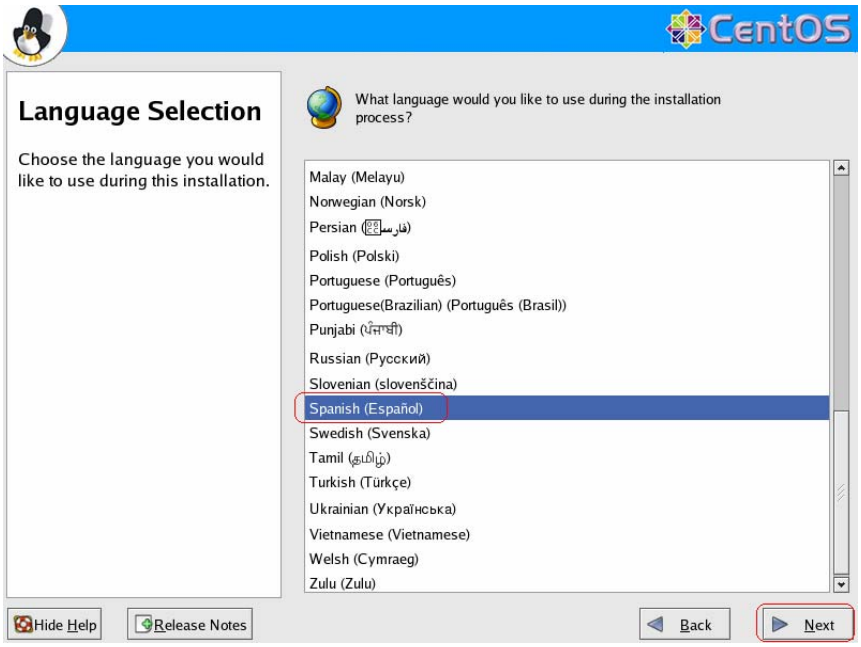


Figura 4-3: Pantalla de selección de lenguaje de instalación

4) Escogemos la configuración adecuada para nuestro teclado en este caso se escogió Spanish y damos click en el botón *Siguiente.*

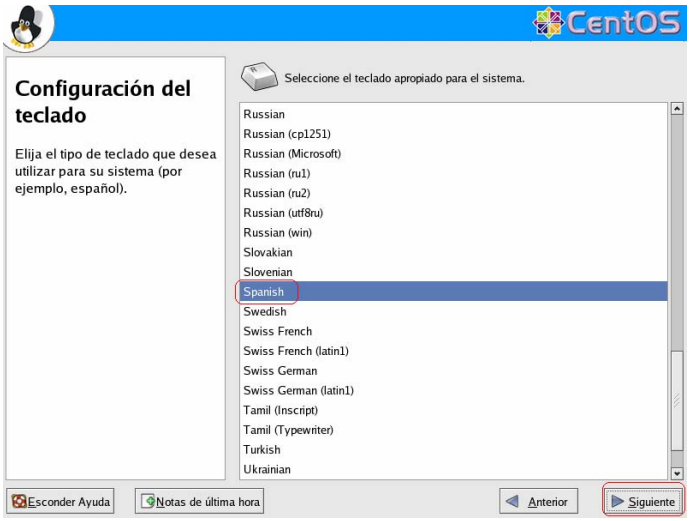


Figura 4-4: Pantalla de selección de lenguaje de instalación

5) Escogemos los paquetes a instalar para el desarrollo de este proyecto se ha escogido la opción personalizada y se ha instalado todos los paquetes.

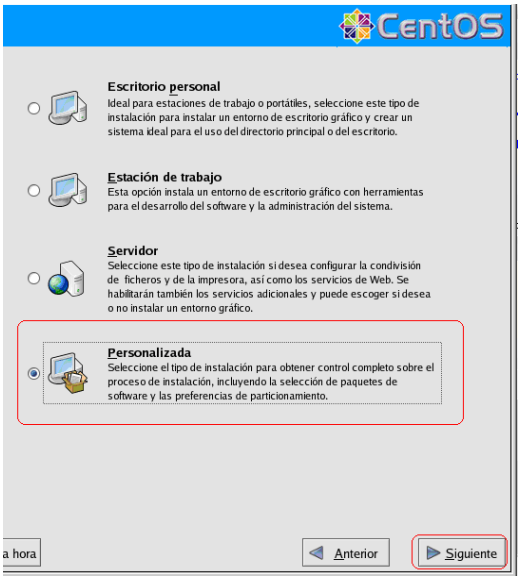


Figura 4-5: Opciones de instalación de CentOS

6) En la configuración del particionamiento del disco duro escogemos la opción Partición manual con *Disk Druid.*

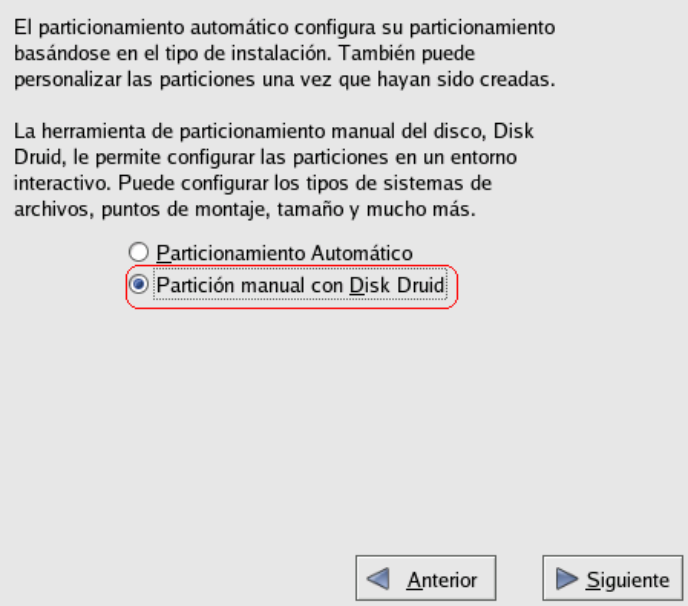


Figura 4-6: Configuración del particionamiento del disco duro

7) Realizar particiones al disco duro para la instalación del sistema operativo CentOS 4.3 (por lo menos dos particiones una tipo */* y una *swap*).

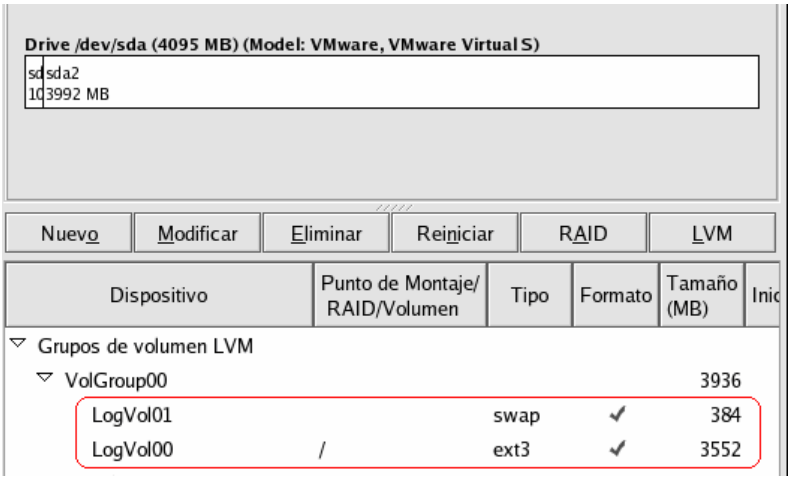


Figura 4-7: Particiones realizadas en el disco duro

8) En la configuración de gestor de arranque dejamos las opciones por defecto dando click en el botón *Siguiente*.

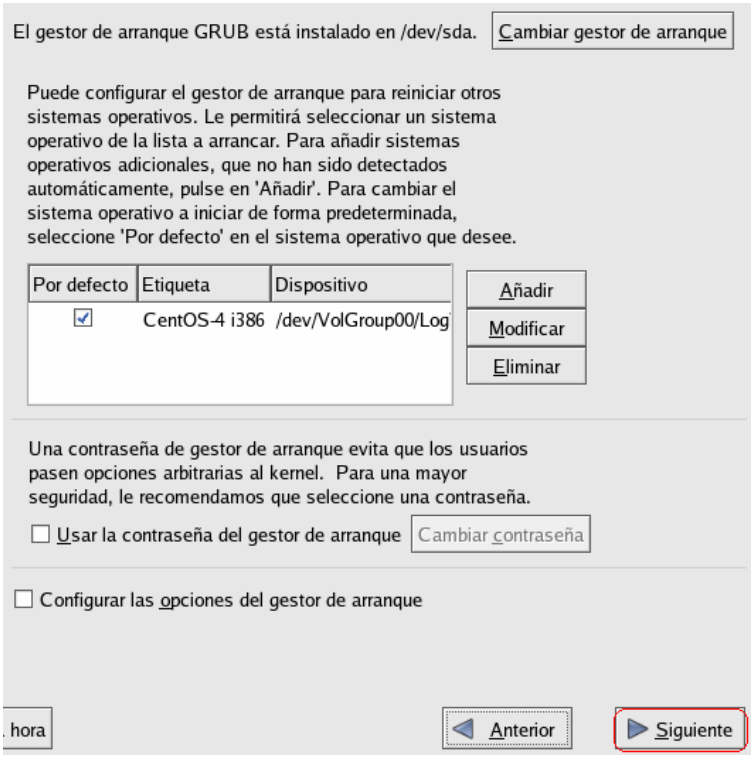


Figura 4-8: Configuración de gestor de arranque

9) En la pantalla de configuración de red realizamos lo siguiente:

- En las opciones de dispositivos de red quitamos en check de *activar al inicio* el dispositivo de red.
- El nombre del host lo ponemos de forma manual y escribimos un nombre en este caso se lo ha nombrado como servidor y damos click en el botón Siguiente.

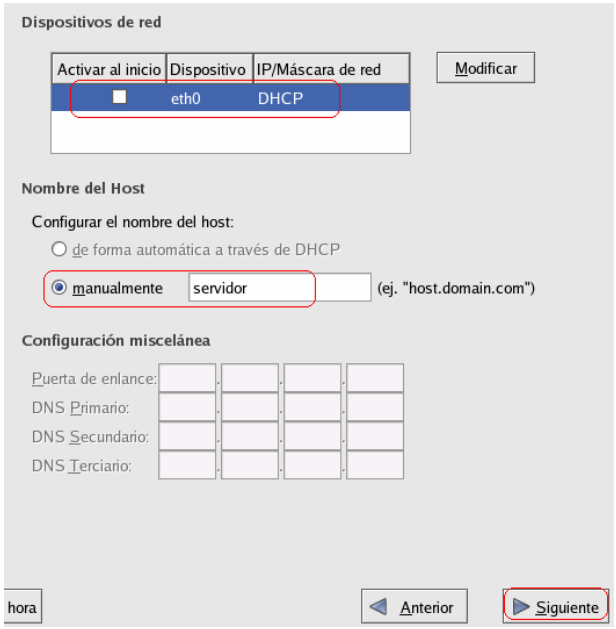


Figura 4-9: Configuración de red

10) En la pantalla de configuración del cortafuegos seleccionamos la opción *Ningún cortafuegos* y damos click en el botón *Siguiente*.

Figura 4-10: Configuración del cortafuegos

11) Cuando nos pida la contraseña de root deberemos tener en cuenta de poner un texto de por lo menos 6 caracteres y confirmarlo nuevamente.

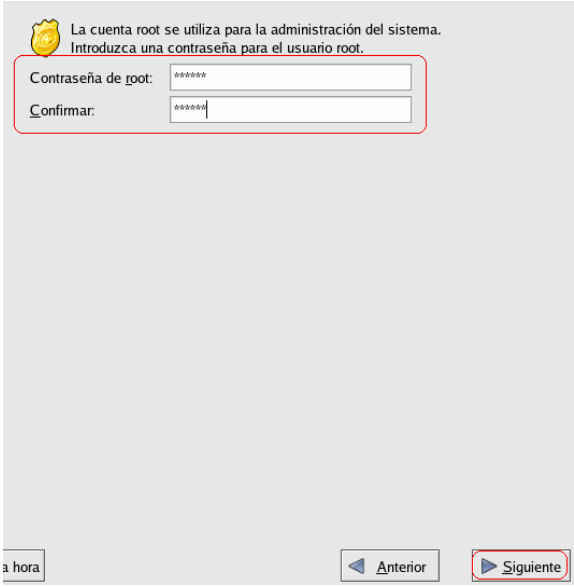


Figura 4-11: Introducción de la contraseña de root

12) Por ultimo en la selección de grupo de paquetes buscamos la *Miscelánea* y seleccionamos ***Todo*** y damos click en el botón *Siguiente*.

Figura 4-12: Selección de grupo de paquetes

4.1.2 Instalación de software de configuración

Webmin es el software que se utilizara como herramienta de configuración en CentOS 4.3 para realizar la instalación de este paquete se ha seguido los siguientes pasos:

1) Bajar el software desde (www.webmin.com).

2) Copiar el archivo bajado en la raíz de nuestro sistema en nuestro caso el archivo tiene el nombre de *webmin-1.300.tar.gz*

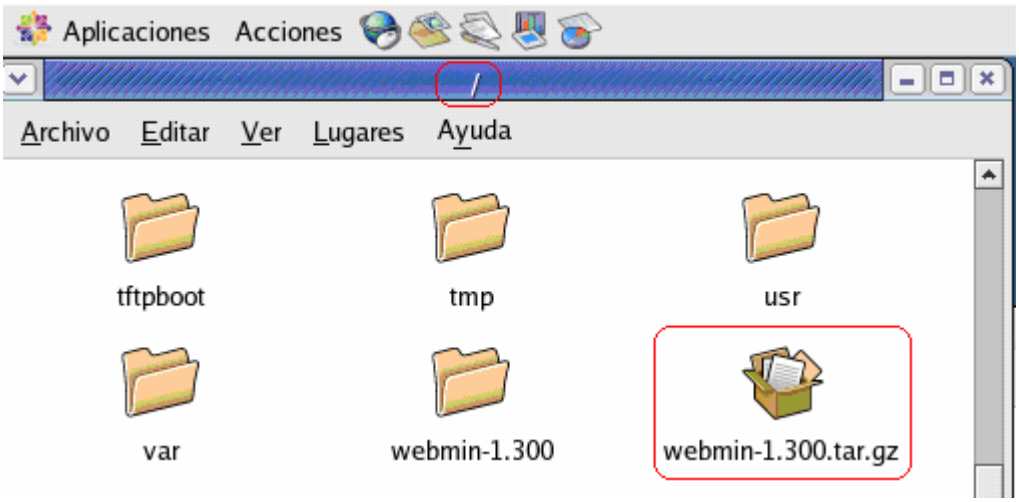


Figura 4-13: Archivo webmin-1.300.tar.gz copiado en la raíz del sistema

3) Ejecutar una ventana de Terminal accediendo por el menú Aplicaciones→ Herramientas del sistema → Terminal.

Figura 4-14: Ejecución de una ventana de Terminal

4) En la ventana de Terminal acceder a la raíz mediante el comando ***cd ..***

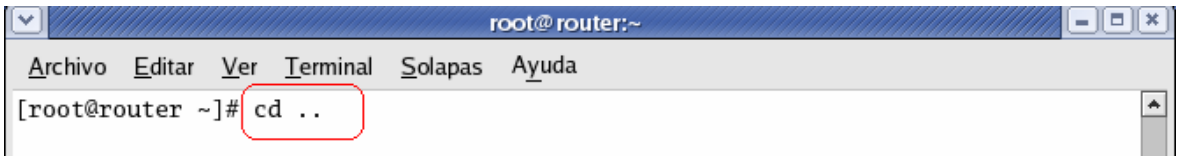


Figura 4-15: Accediendo a la raíz del sistema

5) En la raíz ejecutar el comando **tar xvzf** *webmin-1.300.tar.gz* para descomprimir los archivos en una carpeta con el mismo nombre del archivo original.

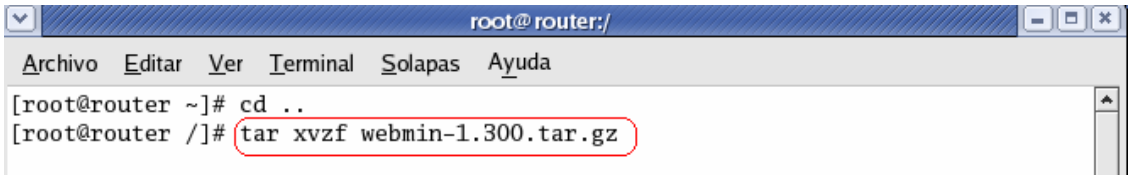


Figura 4-16: Descompresión del archivo webmin-1.300.tar.gz

6) Acceder a la carpeta creada mediante el comando **cd *webmin-1.300***

Figura 4-17: Acceso al directorio creado luego de descomprimir el archivo

7) Ejecutar el comando **./setup.sh** seguido de la dirección donde deseamos que se instalen los archivos en este caso **/usr/local/webmin**

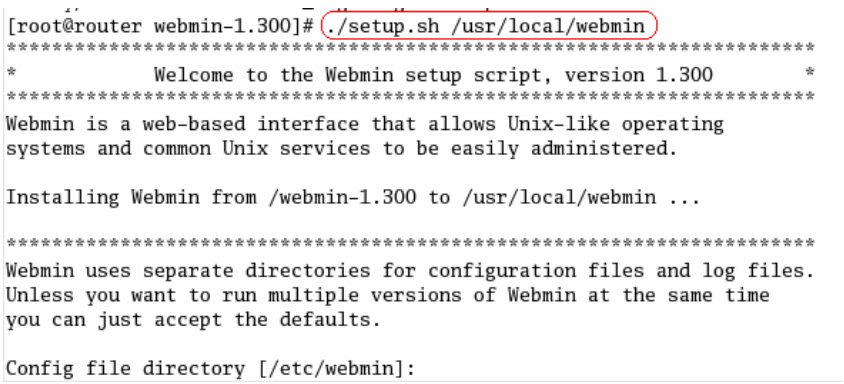


Figura 4-18: Ejecución de la instalación de webmin

8) Al momento que nos pida información sobre el puerto damos un ENTER para dejarlo por defecto en 10000, luego nos pedirá información sobre cual es el usuario administrador allí deberemos colocar root y el password que tiene la cuenta de root.

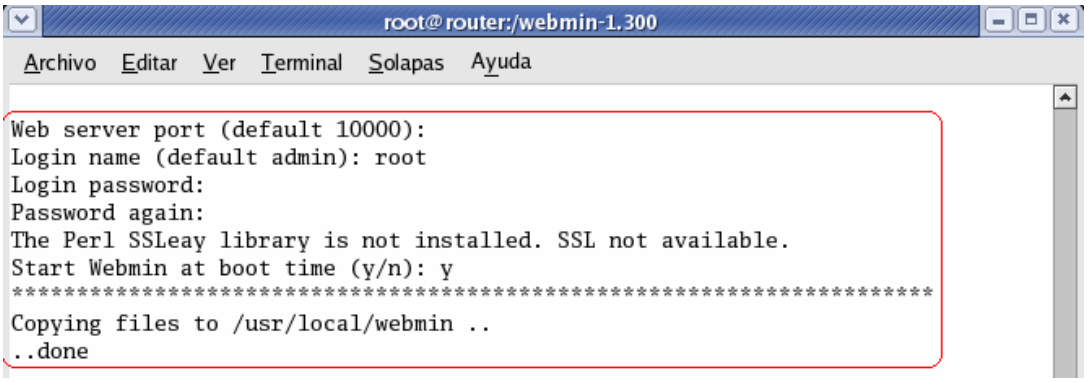


Figura 4-19: Parámetros solicitados durante la instalación de Webmin

9) Finalmente nos dará un mensaje que la instalación ha sido realizada satisfactoriamente y que para utilizarlo deberemos abrir un navegador web cualquiera y teclear http://localhost:10000/ en la barra de direcciones.

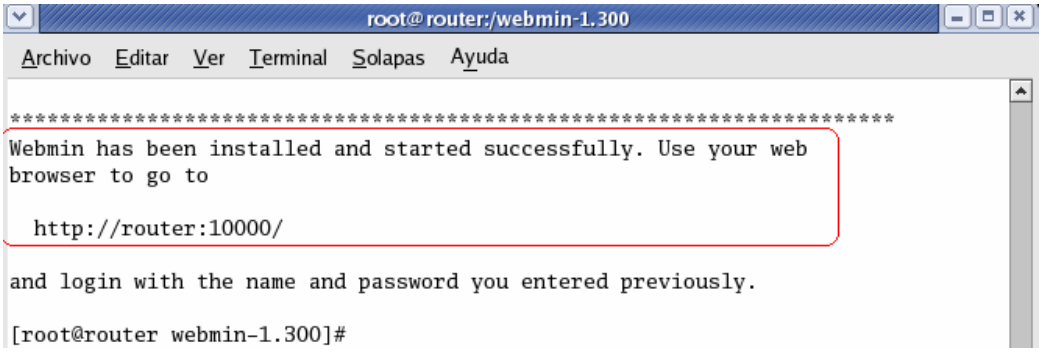


Figura 4-20: Mensaje de finalización de la instalación de Webmin

4.1.3 Instalación de software de ruteo

El software ruteo que se instalo en la computadora que actuara como ruteador es *Zebra* para lo cual se de debe seguir los siguientes pasos para su instalación.

1) Bajar el software desde (www.zebra.org).

2) Copiar el archivo bajado en la raíz de nuestro sistema en nuestro caso el archivo tiene el nombre de *zebra-0.94.tar.gz*

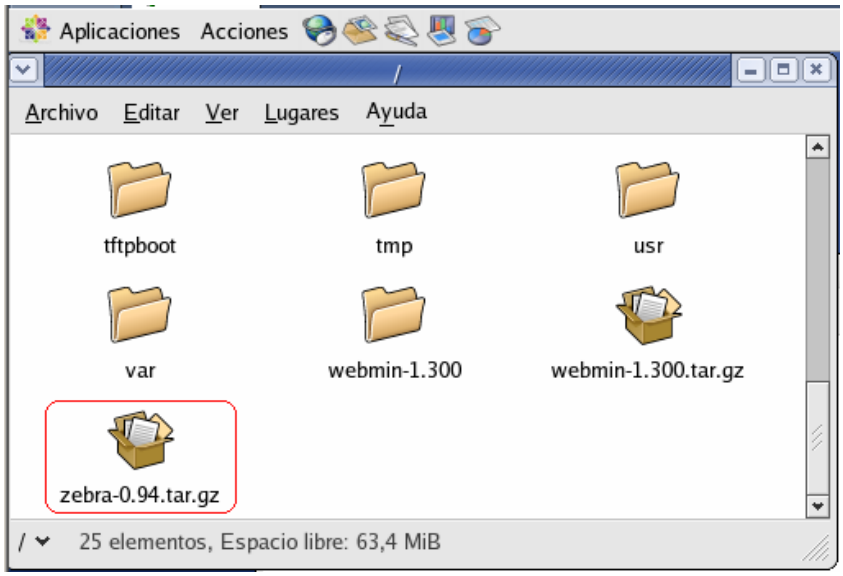


Figura 4-21: Archivo zebra-0.94.tar.gz copiado en la raíz del sistema

3) Ejecutar una ventana de Terminal accediendo por el menú Aplicaciones→ Herramientas del sistema → Terminal.

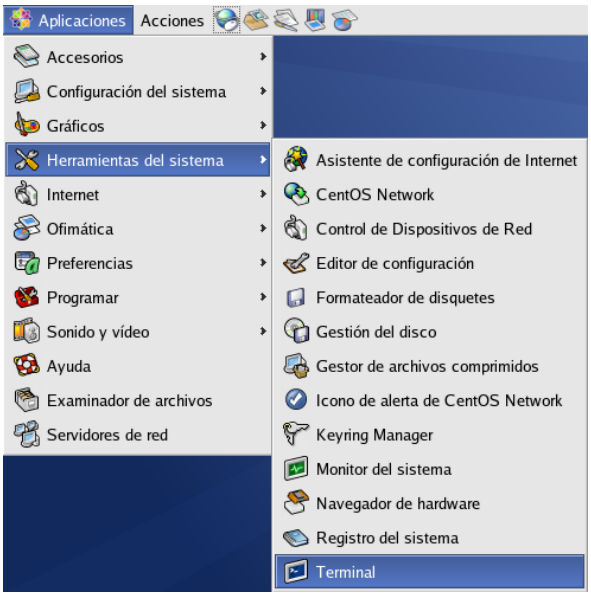


Figura 4-22: Ejecución de una ventana de Terminal

4) En la ventana de Terminal acceder a la raíz mediante el comando **cd ..**

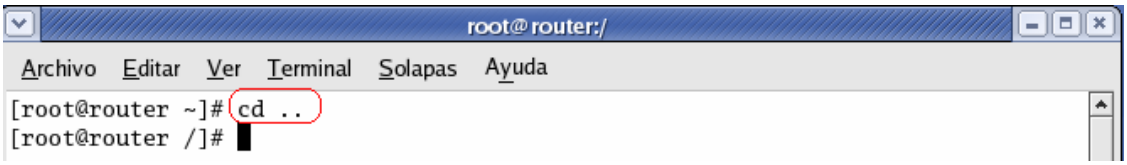


Figura 4-23: Accediendo a la raíz del sistema

5) En la raíz ejecutar el comando **tar xvzf** *zebra-0.94.tar.gz* para descomprimir los archivos en una carpeta con el mismo nombre del archivo original.

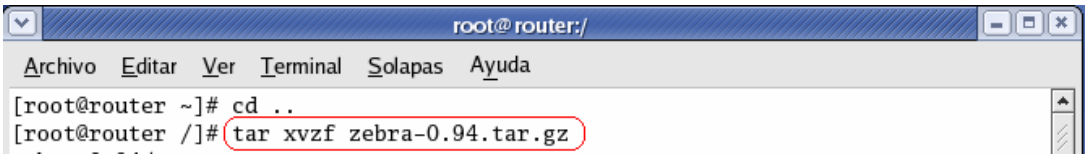


Figura 4-24: Descompresión del archivo zebra-0.94.tar.gz

6) Acceder a la carpeta creada mediante el comando **cd *zebra-0.94***

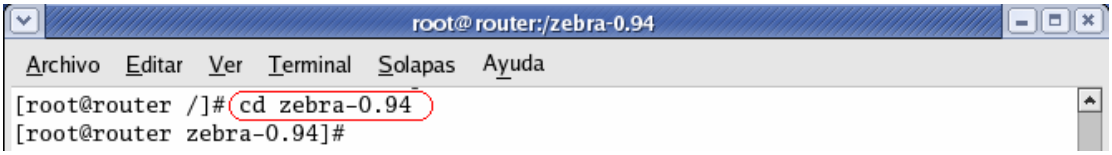


Figura 4-25: Acceso al directorio creado luego de descomprimir el archivo

7) Ejecutar el comando **./configure** y esperar a que termine el proceso.

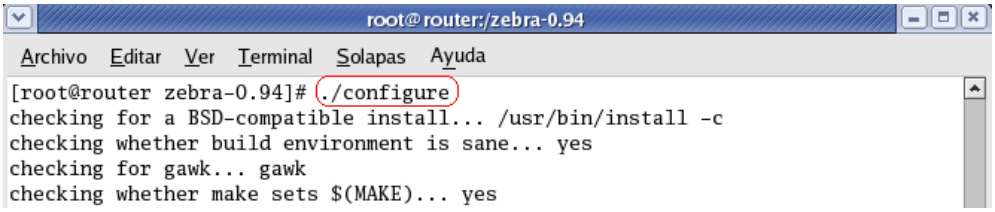


Figura 4-26: Ejecución de la configuración de zebra previa a la instalación

8) Ejecutar el comando **make** y esperar a que termine el proceso.

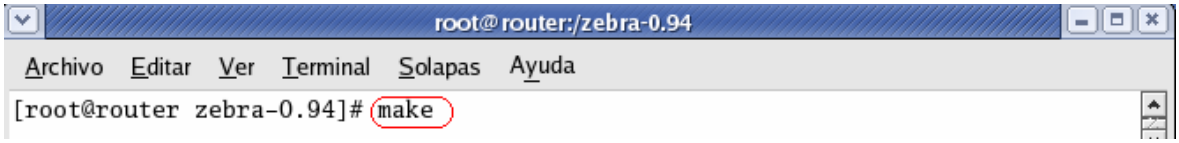


Figura 4-27: Compilación de archivos para la instalación de zebra

9) Opcionalmente se puede ejecutar el comando **make check** para verificar que los archivos creados para la instalación están correctos.

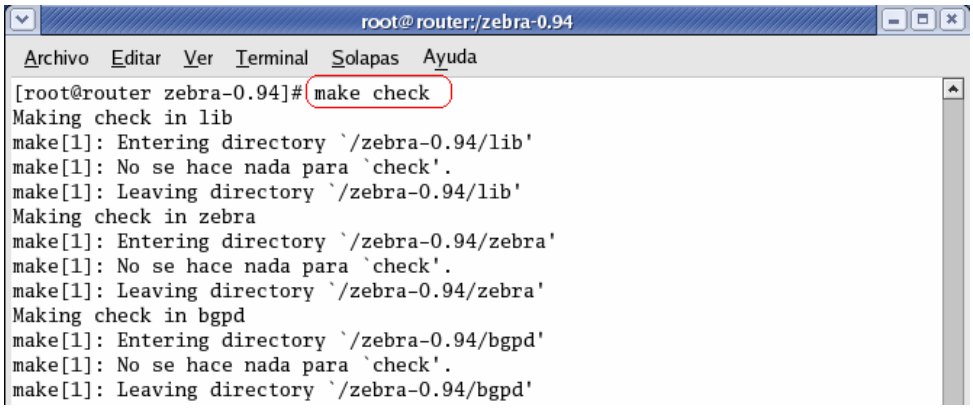


Figura 4-28: Revisión de los archivos compilados

10) Finalmente ejecutar el comando **make install** para realizar la instalación de zebra en el sistema.

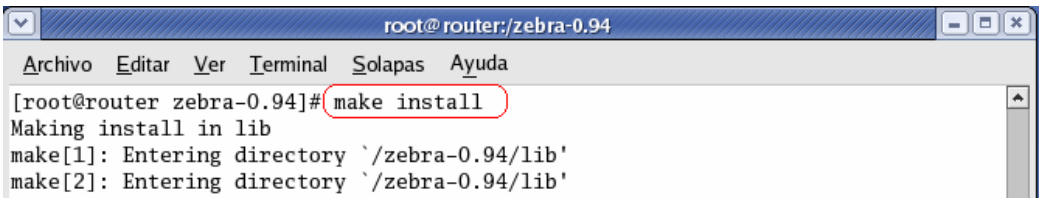


Figura 4-29: Instalación de zebra en el sistema

4.1.4 Instalación de dibbler

El software para montar el servidor DHCP para IPv6 será ***dibbler*** este se lo instalara en la maquina routeadora, para su instalación se debe seguir los siguientes pasos.

1) Bajar el software desde (http://klub.com.pl/dhcpv6/).

2) Copiar el archivo bajado en la raíz de nuestro sistema en nuestro caso el archivo tiene el nombre de *dibbler-0.4.1-src.tar.gz*

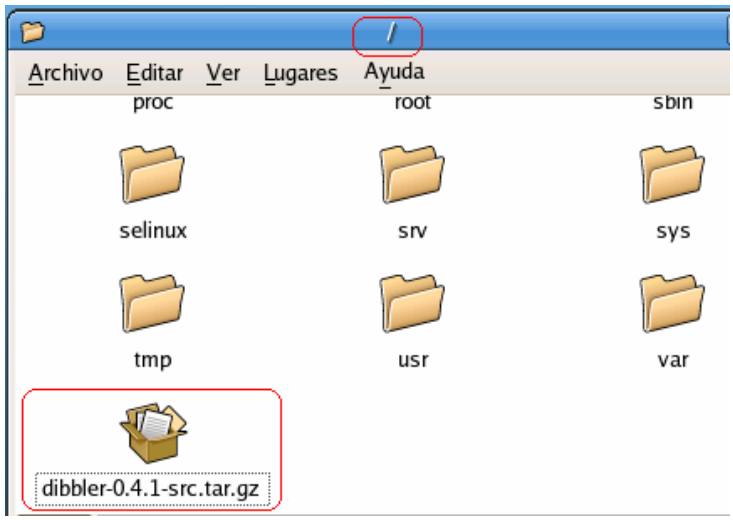


Figura 4-30: Archivo dibbler-0.4.1-src.tar.gz copiado en la raíz del sistema

3) Ejecutar una ventana de Terminal accediendo por el menú Aplicaciones→ Herramientas del sistema → Terminal.

Figura 4-31: Ejecución de una ventana de Terminal

4) En la ventana de Terminal acceder a la raíz mediante el comando ***cd ..***

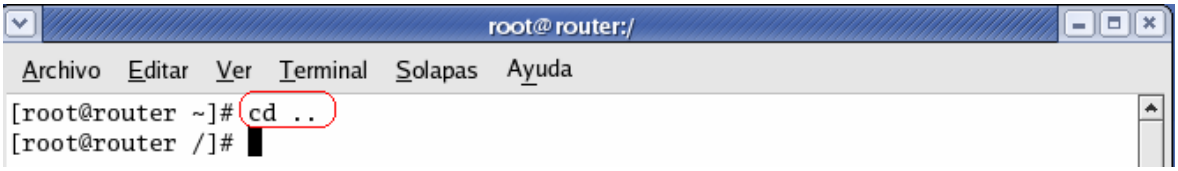


Figura 4-32: Accediendo a la raíz del sistema

5) En la raíz ejecutar el comando ***tar xvzf*** *dibbler-0.4.1.tar.gz* para descomprimir los archivos en una carpeta con el mismo nombre del archivo original.

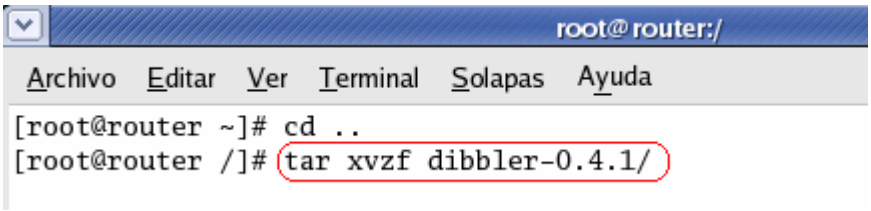


Figura 4-33: Descompresión del archivo dibbler-0.4.1.tar.gz

6) Acceder a la carpeta creada mediante el comando **cd *dibbler-0.4.1***

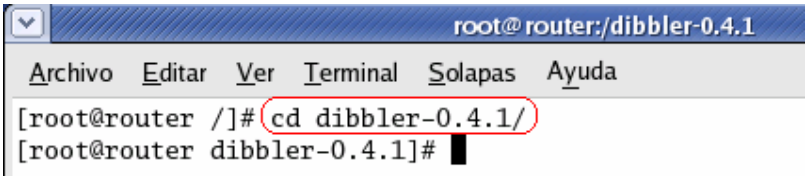


Figura 4-34: Acceso al directorio creado luego de descomprimir el archivo

7) Ejecutar el comando **make server** para realizar una compilación automática de los archivos que se van ha instalar para que cumpla la función de servidor.

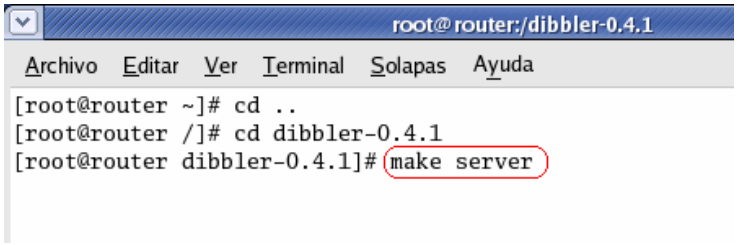


Figura 4-35: Compilador de archivos para la instalación del servidor DHCP

8) Ejecutar el comando ***make install*** para instalar el servidor DHCP dibbler en el sistema.

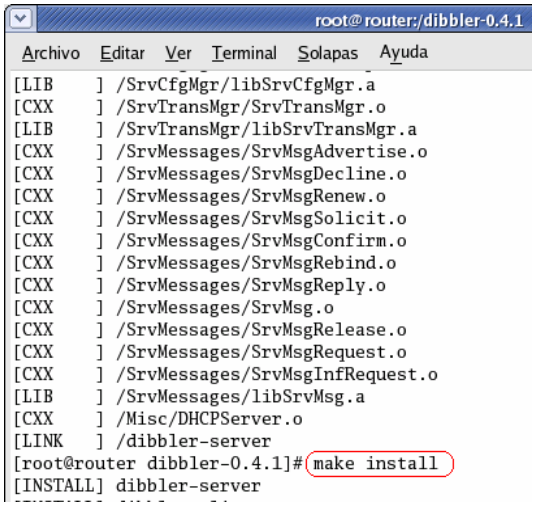


Figura 4-36: Instalación del servidor DHCP en el sistema

9) Para inicializar ó correr el servidor se debe teclear ***dibbler-server start*** en la ventana de Terminal.

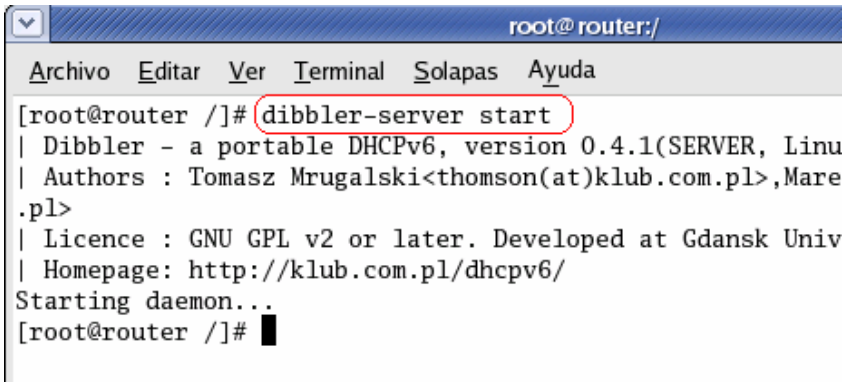


Figura 4-37: Inicialización del servidor DHCP

4.1.5 Instalación del protocolo IPv6

Los sistemas operativos Windows XP SP2, Fedora Core 5 y CentOS 4.3 ya vienen con el protocolo IPv6 preinstalado lo único que hay que hacer para utilizarlo es activarlo. A continuación se presenta el procedimiento realizado en cada sistema operativo para activarlo.

4.1.5.1 Windows XP SP2

Todas las versiones de Windows XP SP2 incluye IPv6 preinstalado pero es preciso habilitarlo. Para habilitarlo es necesario ejecutar con privilegios de administrador el comando ***ipv6 install***.

A continuación se detallan los pasos que se han realizado para habilitarlo en las maquinas clientes

1) Damos click en ***Inicio*** y elegimos ***Ejecutar***

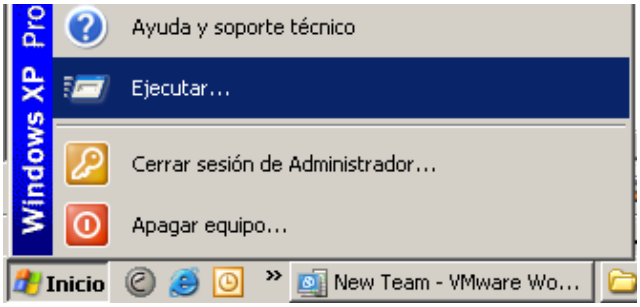


Figura 4-38: Menú de inicio de Windows XP

2) En el casillero de texto tecleamos el comando **cmd** y pulsamos el botón **Aceptar**.

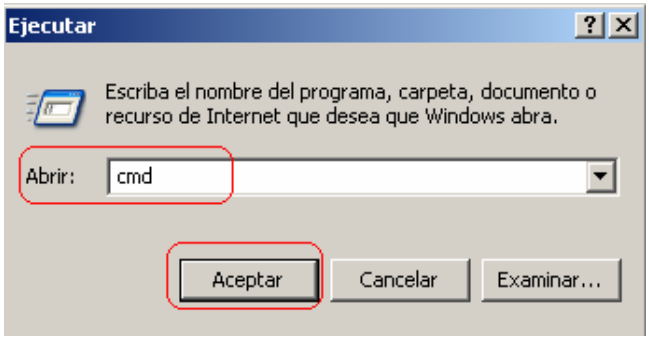


Figura 4-39: Ventana de ejecutar comandos de Windows XP

3) En la ventana de comandos tecleamos ***ipv6 install*** y damos un ***ENTER***.

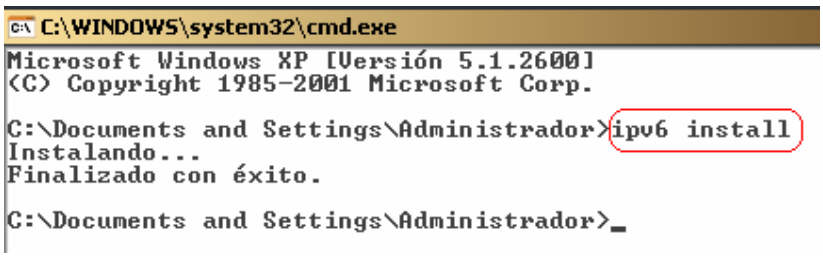


Figura 4-40: Ventana de comandos de Windows XP

4.1.5.2 CentOS 4.3 y Fedora Core 5

En el sistema operativo CentOS 4.3 y Fedora Core 5 ya tienen preinstalado el protocolo IPv6 pero se lo debe activar editando el archivo ***network*** y activando luego la configuración IPv6 para cada una de las interfaces de red que se encuentren instaladas. A continuación se describen los pasos que se hicieron para activar la configuración IPv6.

1) Ir a la carpeta ***/etc/sysconfig*** y buscar el archivo ***network***.

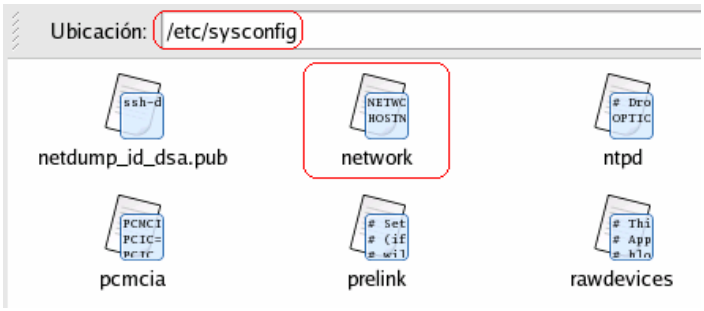


Figura 4-41: Accediendo a la carpeta /etc/sysconfig

2) Abrir el archivo ***network*** ubicado en el ***/etc/sysconfig/network*** con cualquier editor de texto en este caso se esta utilizando gedit y añadir al final del archivo ***NETWORKING_IPV6=yes,*** pulsar el botón Guardar y Salir.

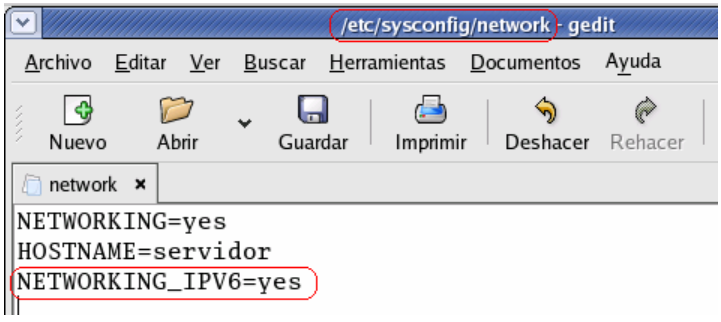


Figura 4-42: Edición del archivo network mediante el editor de texto gedit

3) Ejecutar en una ventana de Terminal el comando *system-config-network-gui*

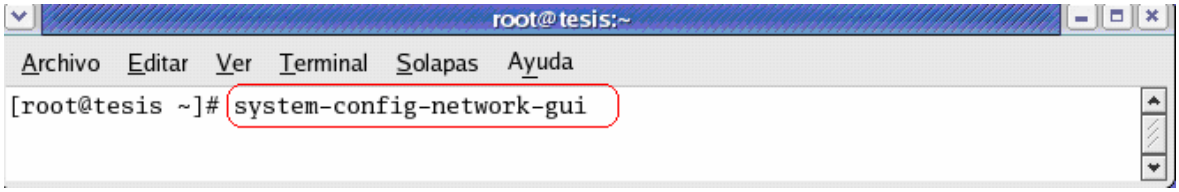


Figura 4-43: Ejecución del comando *system-config-network-gui*

4) Escoger el dispositivo de red y pulsar el botón *Modificar.*

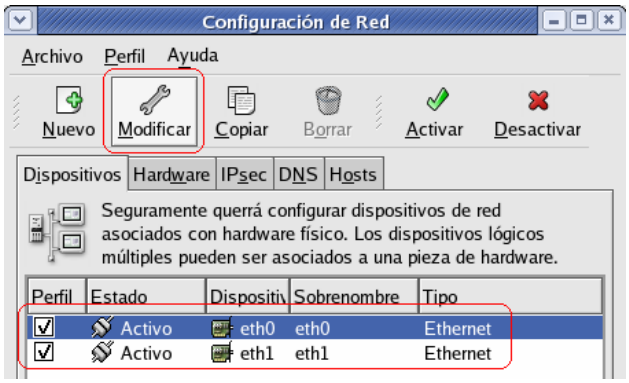


Figura 4-44: Configuración de red en CentOS

5) Señalar la opción *Activar la configuración IPv6 para esta interfaz* y pulsar el botón *Aceptar* para que los cambios sean guardados.

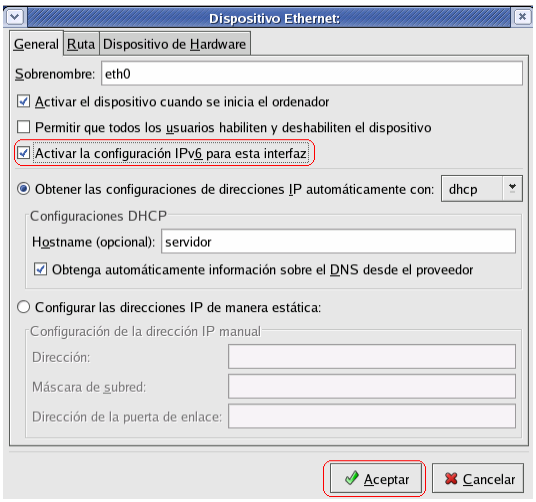


Figura 4-45: Activación de IPv6 en el dispositivo de red

4.2 Configuración de direcciones IPv6

Antes de empezar a trabajar con la intranet de pruebas debemos configurar cada una de las interfases de red de cada computador con una dirección IPv6 y un prefijo, para esto seguiremos el diseño de red mostrado en la Figura 4-46.

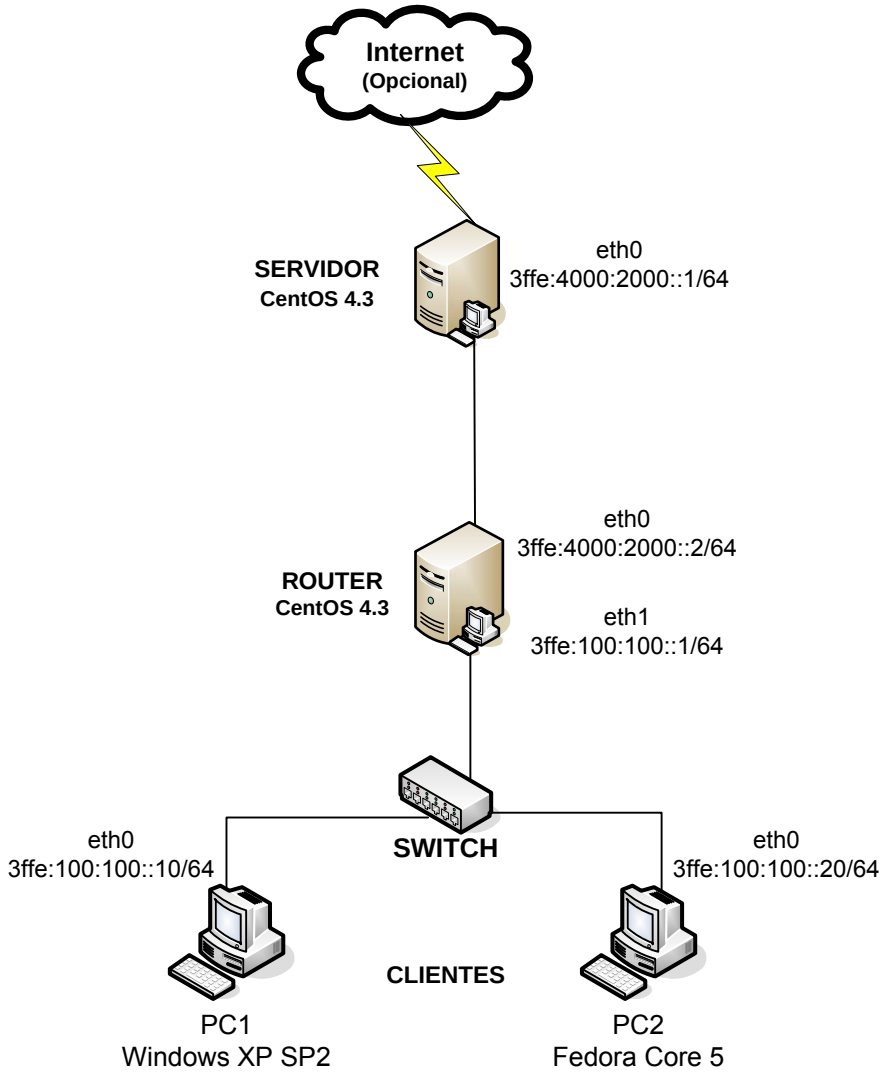


Figura 4-46: Diseño de la red con sus respectivas direcciones IP

La configuración de las direcciones IPv6 asignadas a cada equipo se la ha hecho mediante los comandos que se describen en la Tabla 4-1.

Tabla 4-1: Comandos para configurar direcciones IPv6 manualmente

Sistema operativo	Comando utilizado para añadir direcciones IPv6 a las interfaces de red	Lugar de ejecución
Windows XP SP2	***netsh interface ipv6 add address*** "Nombre de la conexión" Dirección IPv6 ó ***ipv6 adu*** interface/dirección ipv6	Ventana de símbolo de sistema
CentOS 4.3 y Fedora Core 5	***ifconfig eth***X ***add*** Dirección IPv6 / prefijo	Ventana Terminal

4.2.1 Windows XP SP2

Para añadir una dirección IPv6 a una interfase de red en Windows se debe abrir una ventana de símbolo de sistema e introducir el comando ***netsh interface ipv6 add address*** *"Nombre de la conexión" Dirección IPv6*.

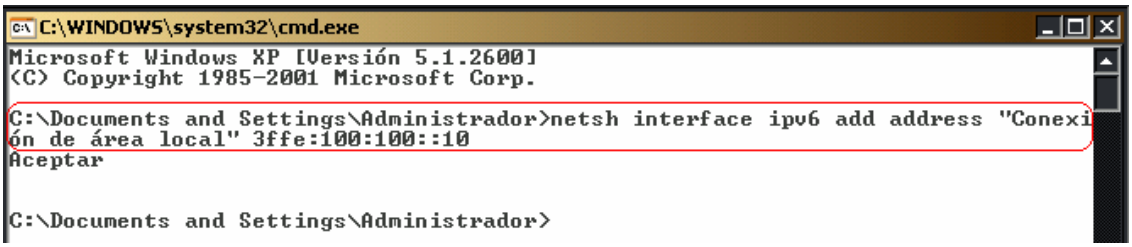


Figura 4-47: Configuración de una dirección IPv6 en Windows XP SP2

4.2.2 CentOS 4.3 y Fedora Core 5

Para añadir una dirección IPv6 a una interface de red en CentOS 4.3 ó Fedora Core 5 se lo puede hacer de dos maneras la una mediante comandos y la otra editando los archivos de configuración.

4.2.2.1 Usando comandos

Para añadir una dirección IPv6 a una interfase de red en Linux se debe abrir una ventana de Terminal e introducir el comando ***ifconfig ethX add*** *Dirección IPv6 / prefijo.*

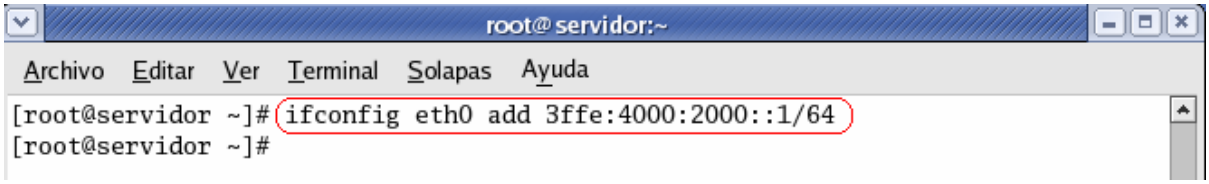


Figura 4-48: Configuración de una dirección IPv6 en CentOS

4.2.2.2 Usando archivos de configuración

La otra forma de añadir una dirección IPv6 en Linux es utilizando los archivos de *network-scripts* el procedimiento para realizarlo es el siguiente:

1) Ir a la carpeta ***/etc/sysconfig/network-scripts*** y abrir el archivo correspondiente a la interfase de red a la que se quiere añadir una dirección

IPv6 en este caso se tiene una solo interfase de red por lo que se tiene un solo archivo asociado llamado *ifcfg-eth0*.

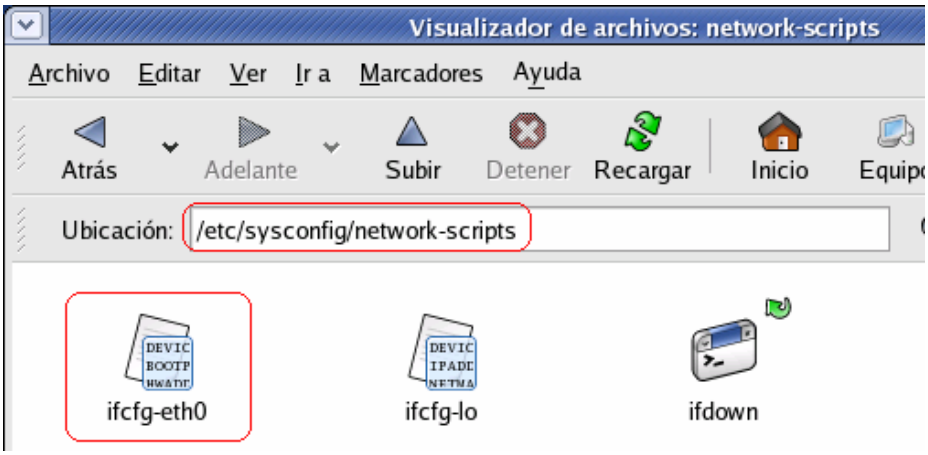


Figura 4-49: Accediendo a la carpeta /etc/sysconfig/network-scripts

2) Abrimos el archivo ***ifcfg-eth0*** con cualquier editor de texto en este caso se ha utilizado el gedit y añadimos la dirección IPv6 tecleando ***IPV6ADDR= dirección_ipv6***.

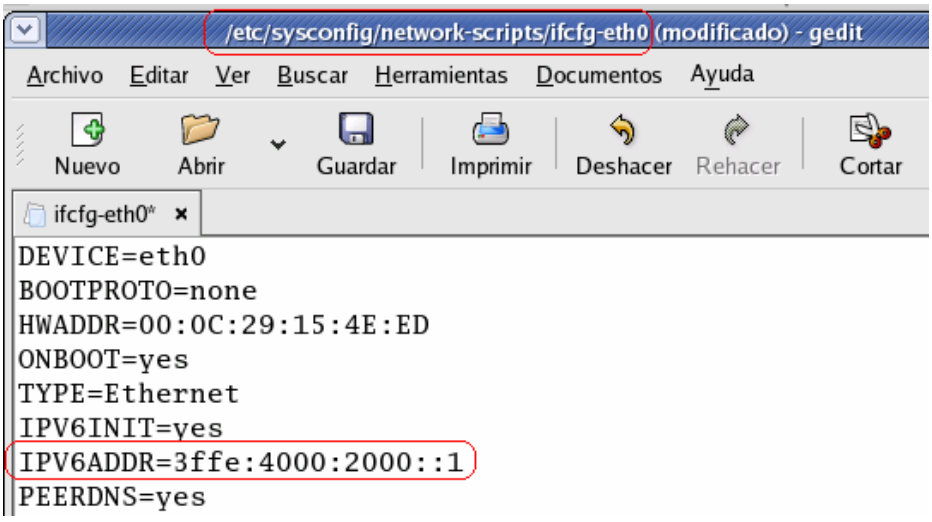


Figura 4-50: Edición del archivo *ifcfg-eth0* mediante el editor de texto gedit

4.3 Configuración de router

La maquina que actuara como router utiliza ***zebra*** como software de ruteo. A continuación se describirá los pasos realizados para la configuración de los archivos tanto de zebra como de ripng.

4.3.1 Zebra

1) Ir a la carpeta ***/usr/local/etc*** y buscar el archivo ***zebra.conf***

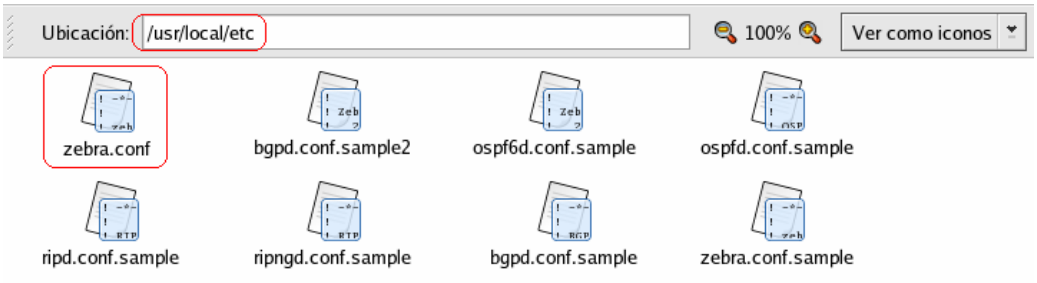


Figura 4-51: Archivo de configuración de zebra

2) Abrimos el archivo ***zebra.conf*** el cual se encuentra en ***/usr/local/etc/*** mediante un editor de textos cualquiera en este caso se lo ha hecho con el ***gedit*** y pondremos el hostname, password para entrar a zebra y el pasword para entrar en el modo privilegiado. Pulsamos Guardar y cerramos el *gedit*.

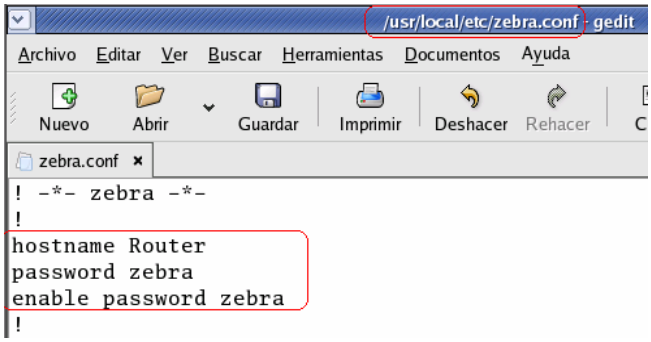


Figura 4-52: Configuración básica del archivo zebra.conf

3) Ejecutar una ventana de Terminal accediendo por el menú Aplicaciones→ Herramientas del sistema → Terminal.

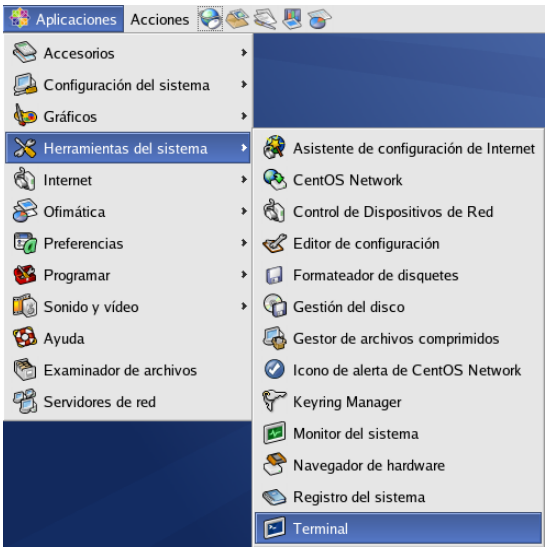


Figura 4-53: Ejecución de una ventana de Terminal

4) En la ventana de Terminal tecleamos **zebra –d** para levantar el zebra y luego ***telnet localhost zebra*** para acceder a la configuración de zebra nos pedirá un password en este caso el password es zebra y damos un *Enter*.

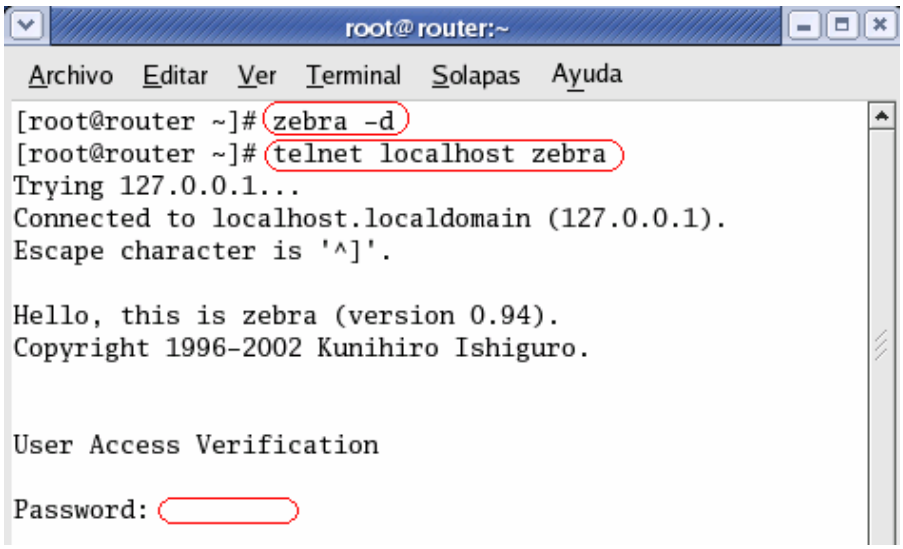


Figura 4-54: Acceso a zebra mediante telnet

5) Una vez dentro de zebra habilitamos el modo privilegiado mediante el comando ***enable*** y nos pedirá el pasword para entrar a este modo aquí teclearemos zebra.

6) Una vez dentro del modo privilegiado estableceremos el modo de configuración del enrutador mediante el comando **configure terminal**.

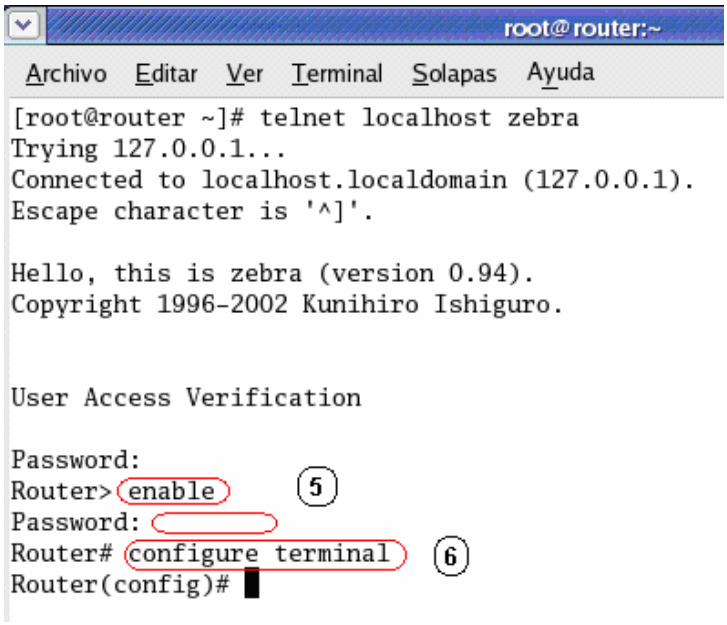


Figura 4-55: Acceso al modo privilegiado

7) En el modo de configuración del enrutador se ha configurado las direcciones IPv6 para cada interface de red, esto se lo realiza indicando la interface de red a la cual se quiere añadir una dirección mediante el comando ***interface ethx*** y luego mediante el comando ***ipv6 address*** *direccion_ipv6* se añade la dirección IP para la interface de red seleccionada.

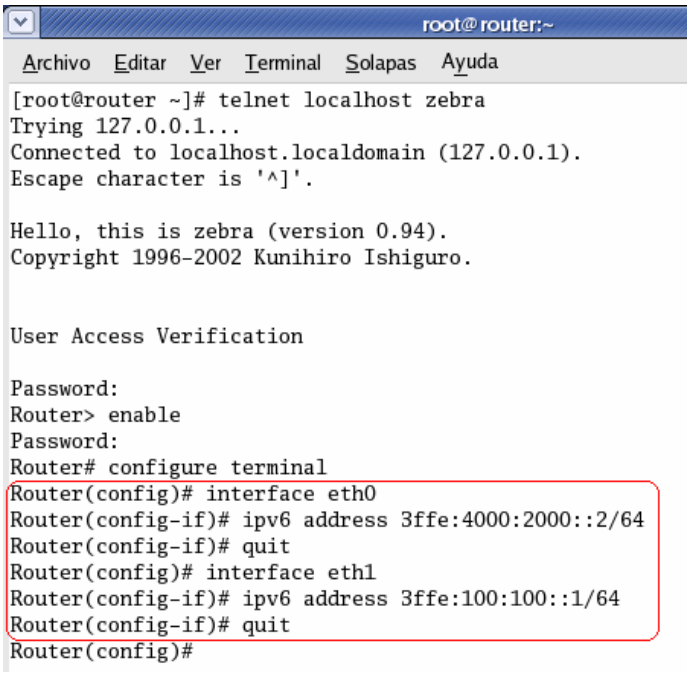


Figura 4-56: Configuración de las interfaces del router

8) Finalmente guardamos la configuración mediante el comando ***write*** el cual nos da un aviso que la configuración fue guardada en el archivo ***/usr/local/etc/zebra.conf***.

Figura 4-57: Guardando la configuración de zebra

4.3.2 Ripng

1) Ir a la carpeta ***/usr/local/etc*** y buscar el archivo ***ripngd.conf***.

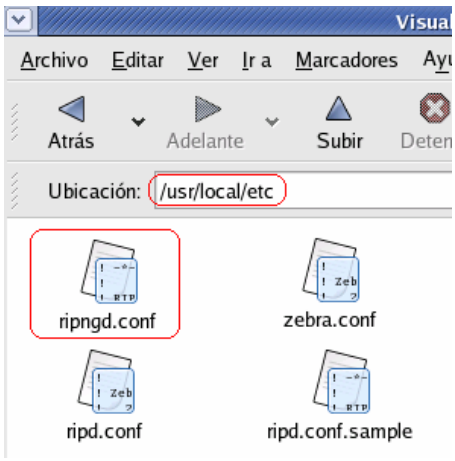


Figura 4-58: Archivo de configuración de ripng

2) Abrimos el archivo ***ripngd.conf*** el cual se encuentra en ***/usr/local/etc/*** mediante un editor de textos cualquiera en este caso se lo ha hecho con el ***gedit*** y pondremos el hostname, password para entrar a ripng y el pasword para entrar en el modo privilegiado. Pulsamos Guardar y cerramos el *gedit*.

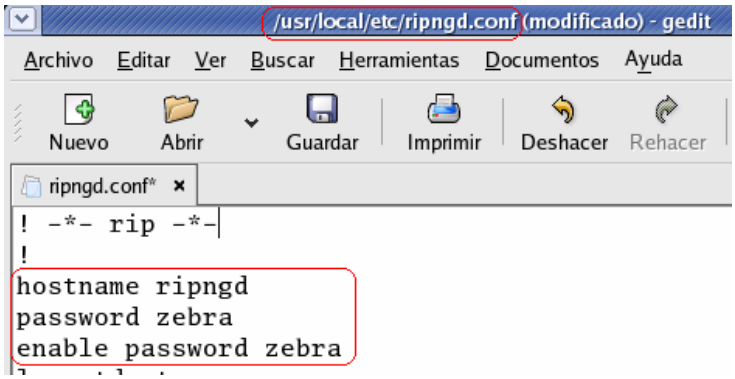


Figura 4-59: Configuración básica del archivo ripngd.conf

3) Ejecutar una ventana de Terminal accediendo por el menú Aplicaciones→ Herramientas del sistema → Terminal.

Figura 4-60: Ejecución de una ventana de Terminal

4) En la ventana de Terminal tecleamos ***ripngd –d*** para arrancar el demonio de ripng, antes de levantar ripngd se debe verificar que el zebra este levantado.

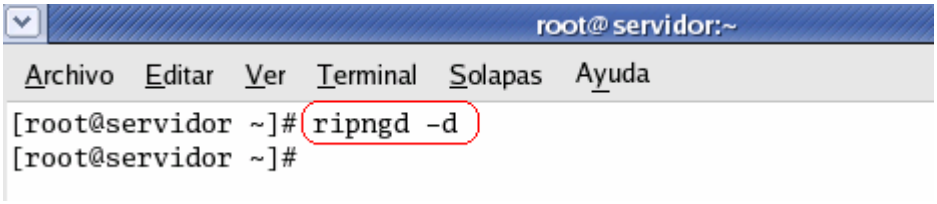


Figura 4-61: Arrancando ripng en una ventana de Terminal

5) En la ventana de Terminal teclear ***telnet localhost ripngd*** para acceder a la configuración de ripng nos pedirá un password en este caso el password es zebra y damos un Enter.

Figura 4-62: Acceso a ripng mediante telnet

6) Una vez dentro de ripng habilitamos el modo privilegiado mediante el comando ***enable*** y nos pedirá el pasword para entrar a este modo aquí tecleamos zebra.

7) Una vez dentro del modo privilegiado estableceremos el modo de configuración del enrutador mediante el comando **configure terminal**.

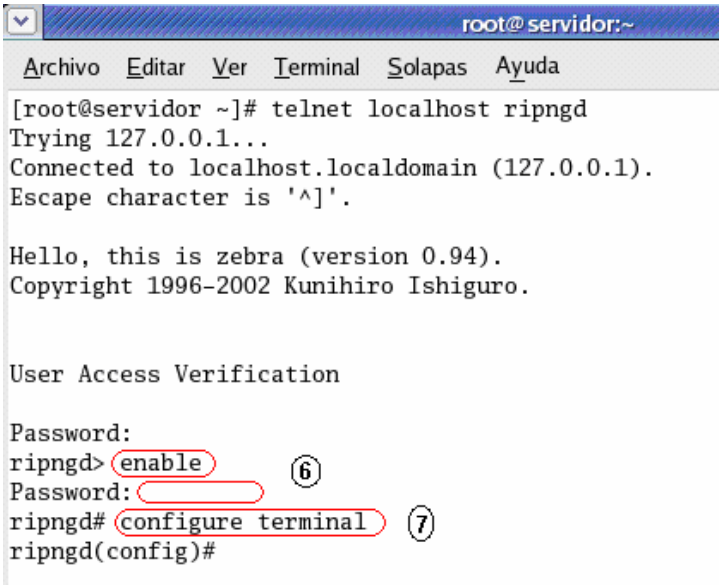


Figura 4-63: Acceso al modo privilegiado

8) En el modo de configuración del enrutador se ha realizado lo siguiente:

- La habilitación del protocolo ripng mediante el comando ***route ripng***
- Especificación de las redes directamente conectadas al equipo por las cuales queremos que el router enrute con el protocolo ripng esto se lo realiza mediante el comando ***network*** en este caso se ha especificado que coja la dirección de red que tenga tanto la interface *eth0* como la *eth1*.
- Y finalmente se indica que el intercambio de información de enrutamiento se lo haga mediante las interfaces directamente conectadas esto se lo hace mediante el comando ***redistribute connected***.

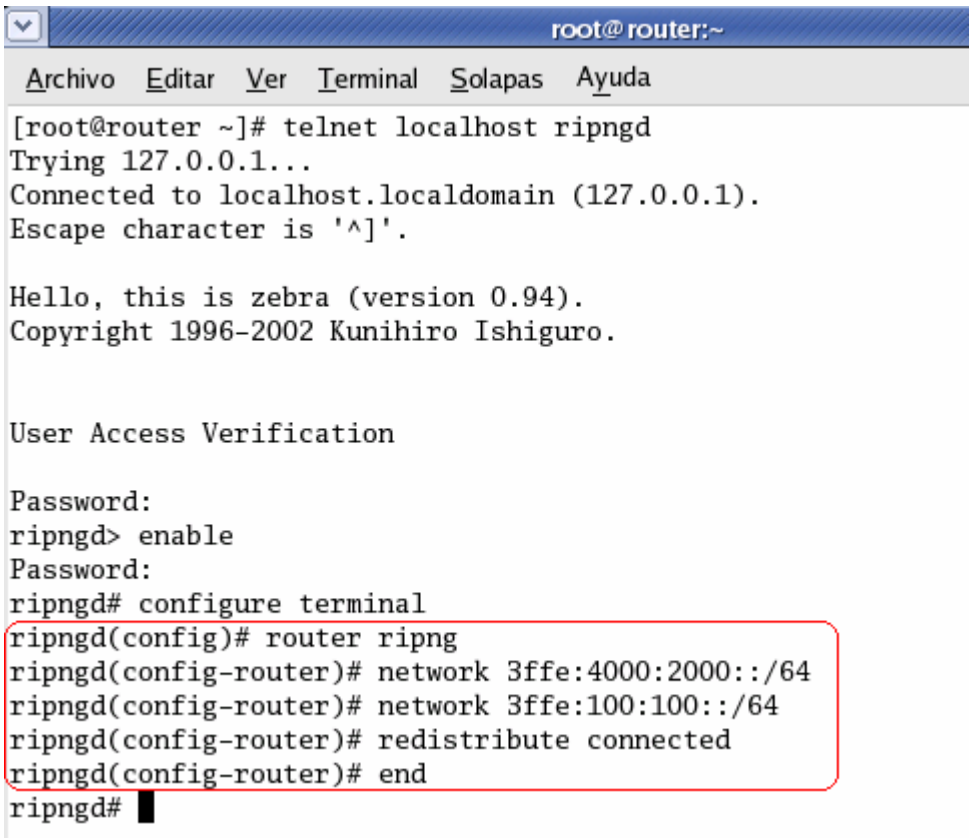


Figura 4-64: Configuración de ripng

9) Finalmente guardamos la configuración mediante el comando ***write*** el cual nos da un aviso que la configuración fue guardada en el archivo ***/usr/local/etc/ripngd.conf***

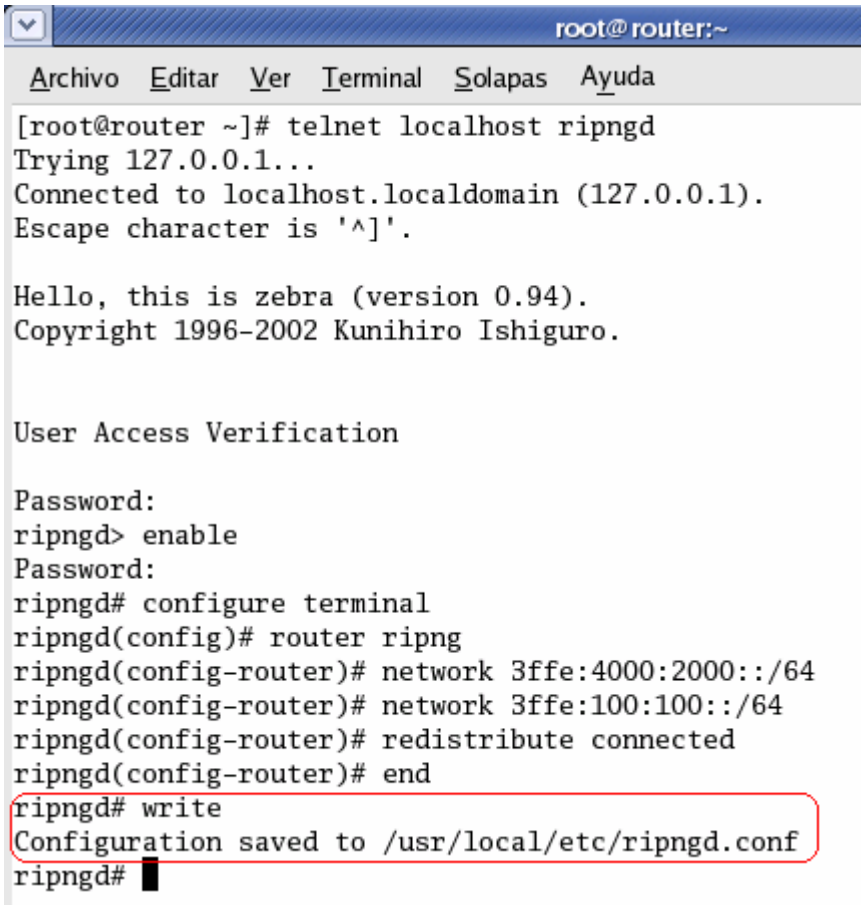


Figura 4-65: Guardando la configuración de ripng

4.4 Configuración de Servidores

Para la configuración de los servidores de red y/o servicios de Internet se han utilizado diferentes paquetes unos incluidos en la distribución de CentOS 4.3 (Tabla 4-2) y otros ajenos a esta distribución de sistema operativo; estos paquetes han sido configurados utilizando los archivos de configuración mediante la herramienta Webmin ó utilizando un editor de texto para modificar los archivos de configuración de cada uno de estos paquetes.

Tabla 4-2: Paquetes utilizados para levantar los diferentes servidores

Servidor	**Paquete utilizado**	**Incluido en la distribución de CentOS**	**Comando a ejecutar en una ventana de Terminal para arrancar el servicio**	**Dirección IP**
DNS	BIND	Si	/etc/init.d/named start	3ffe:4000:2000::1
DHCP	Dibbler	No	dibbler-server start	3ffe:100:100::1
SMTP	Sendmail	Si	/etc/init.d/sendmail start	3ffe:4000:2000::1
POP3	Dovecot	Si	/etc/init.d/dovecot start	3ffe:4000:2000::1
Web	Apache	Si	/etc/init.d/httpd start	3ffe:4000:2000::1
FTP	Vsftpd	Si	/etc/init.d/vsftpd start	3ffe:4000:2000::1
SSH	Sshd	Si	/etc/init.d/sshd start	3ffe:4000:2000::1

4.4.1 Servidor DNS

Como servidor de DNS se ha utilizado BIND incluido en la distribución de CentOS, para realizar la configuración de BIND utilizando la herramienta de configuración Webmin para lo cual se ha realizado el siguiente procedimiento:

1) Abrimos un navegador Web cualquiera en nuestro caso se esta utilizando Mozilla Firefox y tecleamos en la barra de direcciones http://localhost:10000

2) Para acceder deberemos poner el username y pasword del administrador para nuestro proyecto el servidor tiene el username : root y el pasword : servidor

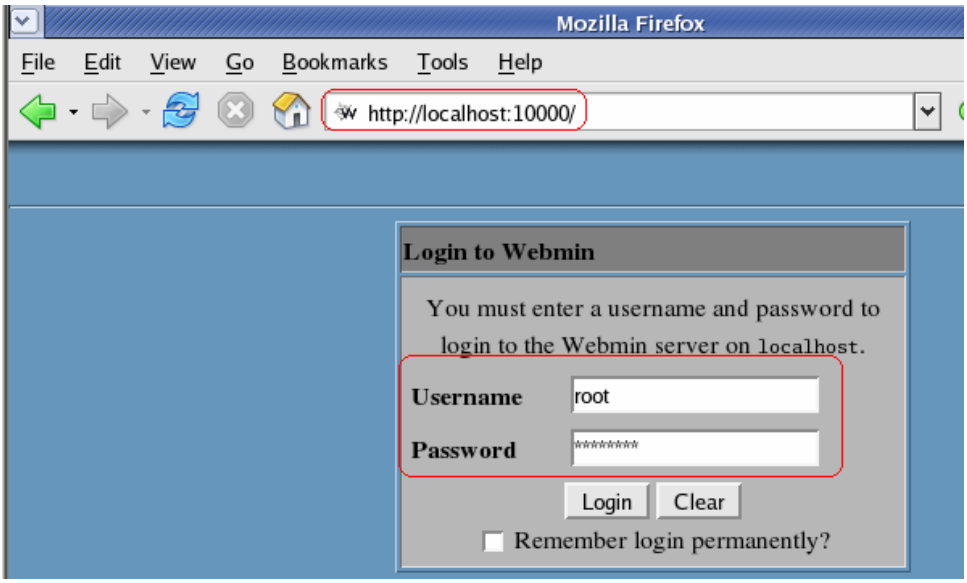


Figura 4-66: Pantalla de autentificación de Webmin

3) Hacemos click en la pestaña de Servidores.

4) Damos click en la opción con texto: Servidor de DNS BIND.

Figura 4-67: Icono de acceso a la configuración de BIND en Webmin

5) Buscamos la opción ***Edit Config File*** y damos un click para acceder.

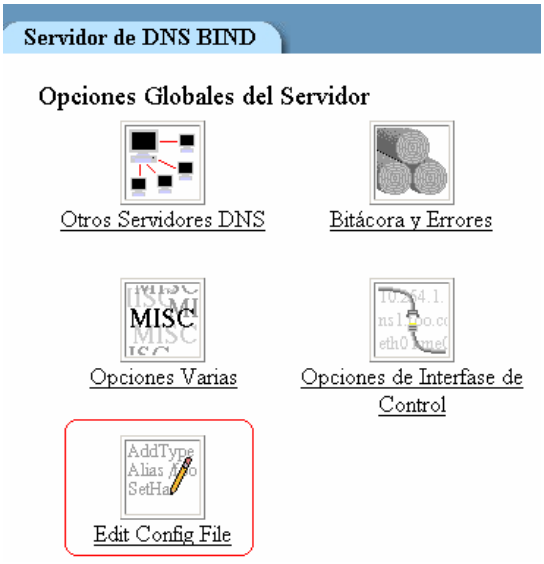


Figura 4-68: Opciones globales del servidor BIND

6) Elegimos el archivo de configuración ubicado en ***/etc/named.conf*** y pulsamos el botón ***Edit***.

7) En el archivo ***/etc/named.conf*** se ha añadido la siguiente línea de texto: ***listen-on-v6{any;}*** para poder escuchar peticiones de resolución de direcciones IPv6 y pulsamos el botón Salvar para guardar los cambios.

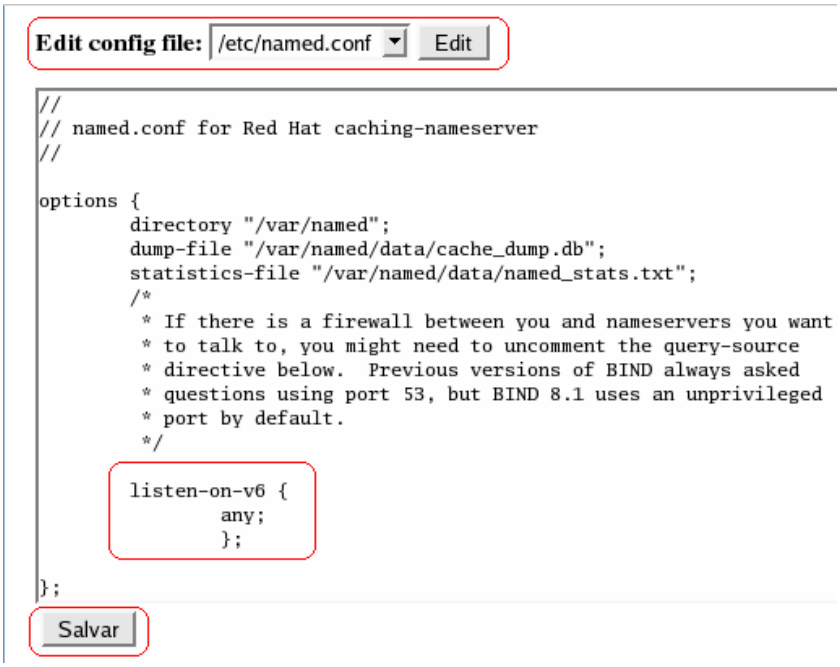


Figura 4-69: Configuración de archivos del servidor DNS

4.4.1.1 Creación de zona de reenvió

1) Creamos una zona maestra pulsando sobre el texto ***Crear una nueva zona maestra***.

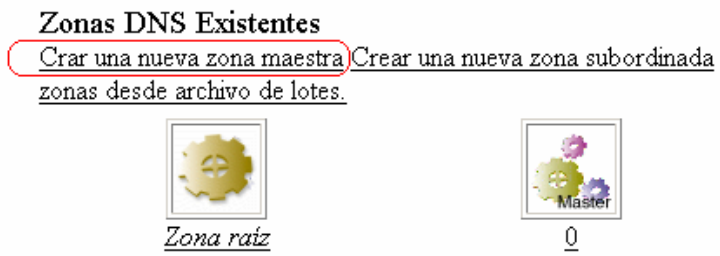


Figura 4-70: Creación de zonas del servidor DNS

2) En las opciones de la nueva zona maestra llenamos:

- En Tipo de zona escogemos la opción de Reenvió
- El nombre de dominio (para este proyecto es ***ipv6.pro***),
- La dirección de correo del administrador de red (javier@ipv6.pro),
- y pulsamos el botón Crear

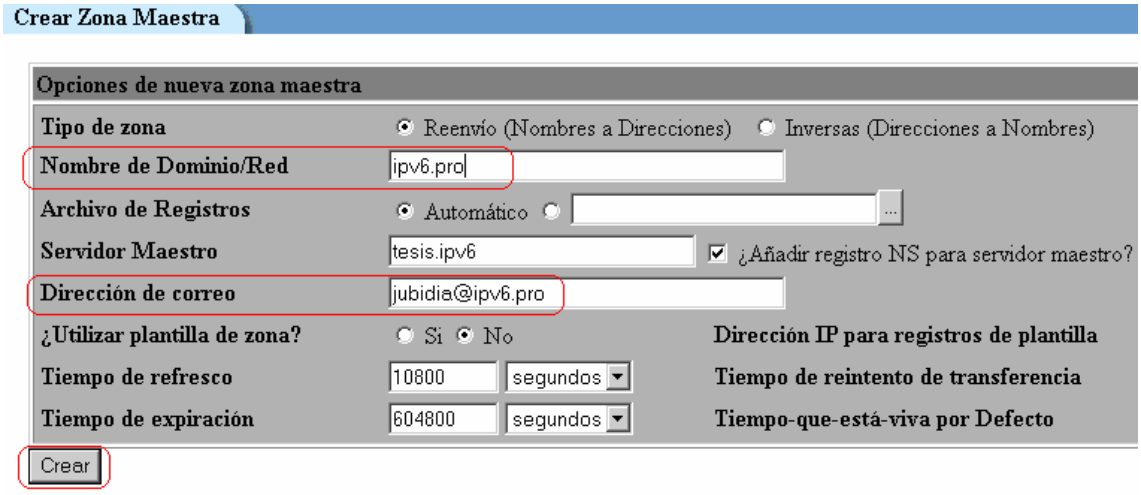


Figura 4-71: Opciones de la zona a crear

3) En el índice del modulo en zonas existentes buscamos la zona que creamos en este caso ***ipv6.pro*** y damos un click para entrar en las opciones de Editar Zona Maestra.

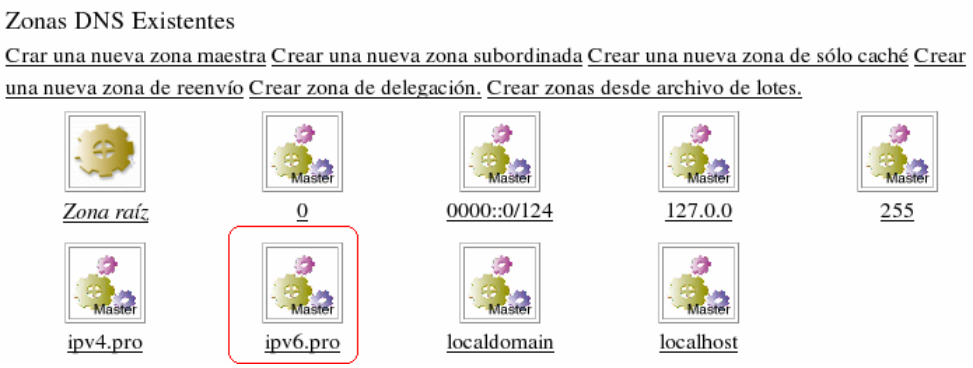


Figura 4-72: Zona DNS creada

4) Ya en la opción de Editar Zona Maestra buscamos la opción de ***Editar Archivos de Registro*** y damos un click para acceder.

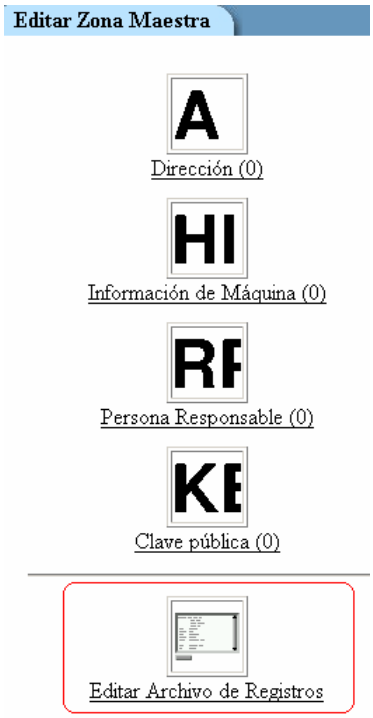


Figura 4-73: Edición de archivos de registro

5) Añadimos los registros DNS de tipo AAAA en el archivo ubicado en **/var/named/ipv6.pro.hosts** para los servicios levantados en nuestro servidor y pulsamos en botón ***Salvar***

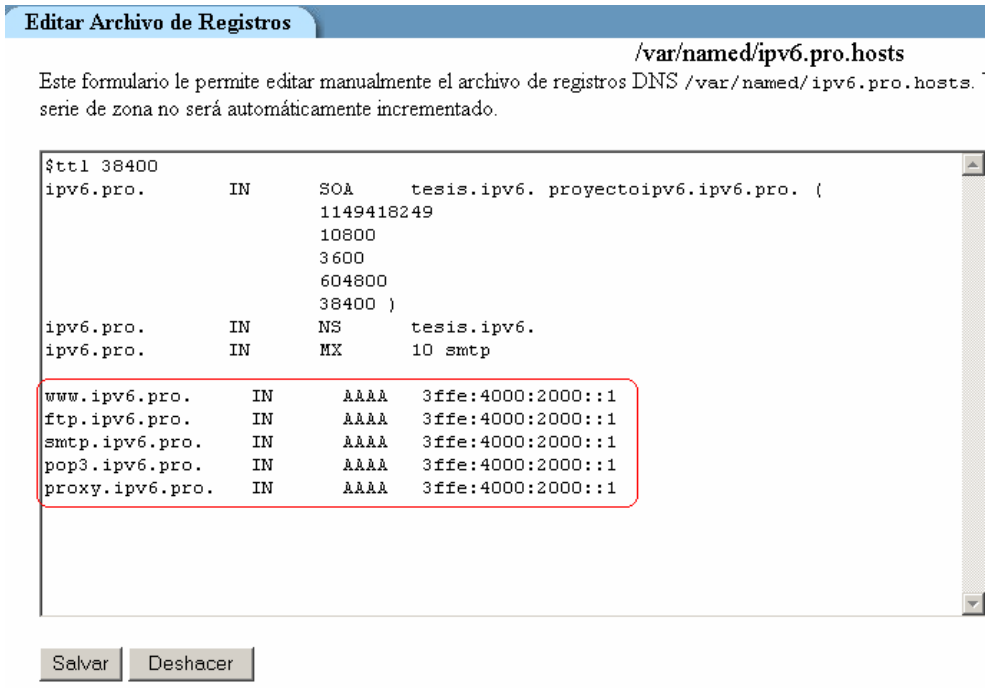


Figura 4-74: Líneas de texto introducidas en el archivo de registro

4.4.1.2 Creación de zona inversa

1) Editamos el archivo ***named.conf*** ubicado en ***/etc/named.conf*** en el cual pondremos la dirección de red IPv6 3ffe:4000:2000::0/64 en forma inversa separado por puntos y completando los 64 bits del prefijo y en la terminación pondremos ***ip6.arpa*** para indicar que se trata de una dirección IPv6.

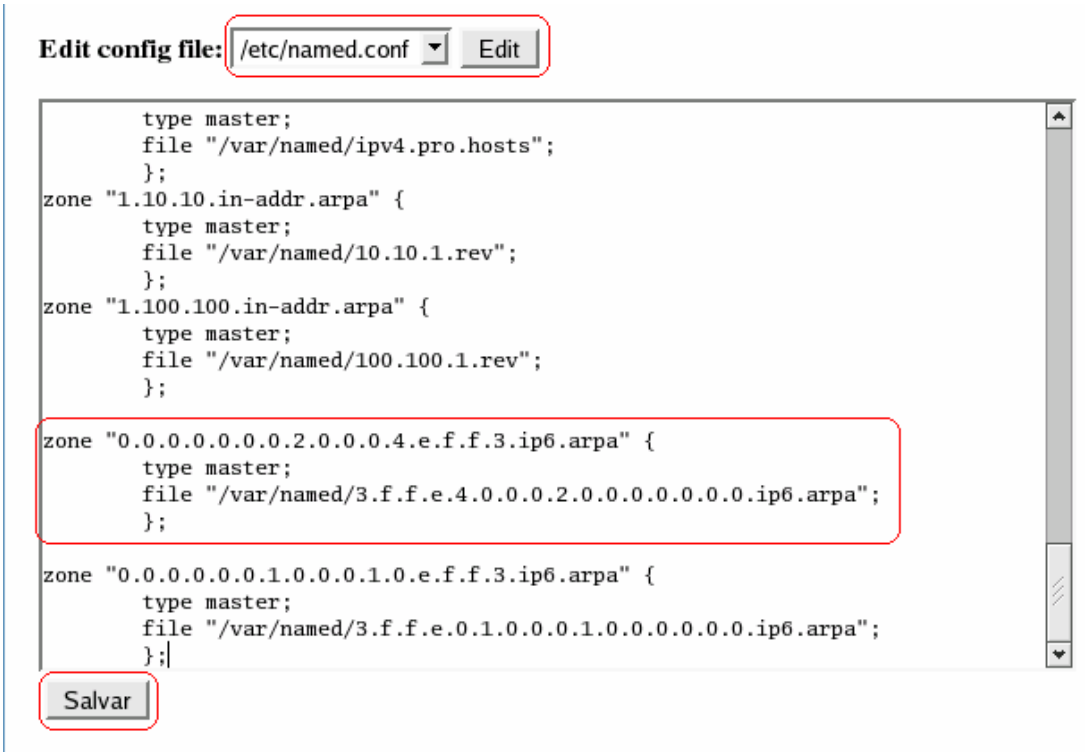


Figura 4-75: Creación de zona inversa en el archivo named.conf

2) En el índice del modulo en zonas existentes buscamos la zona que creamos en este caso ***3ffe:4000:2000:0::/64*** y damos un click para entrar en las opciones de Editar Zona Maestra.

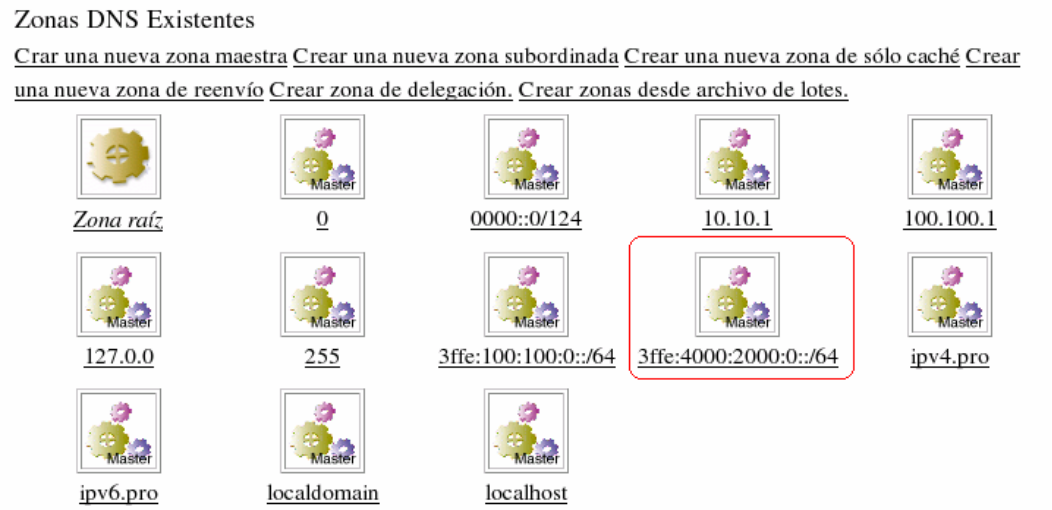


Figura 4-76: Zona DNS inversa

3) Ya en la opción de Editar Zona Maestra buscamos la opción de ***Editar Archivos de Registro*** y damos un click para acceder.

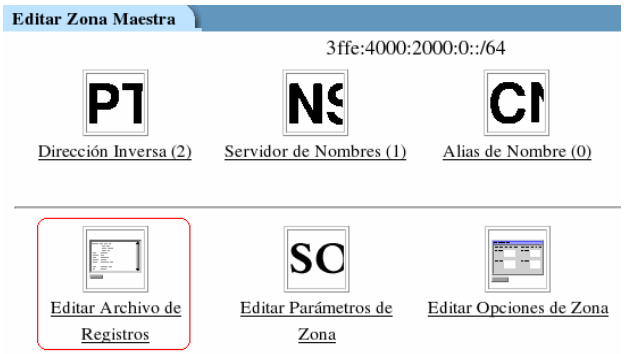


Figura 4-77: Opción de edición de archivos de registros

4) Añadimos los registros DNS de tipo PTR en el archivo ubicado en **/var/named/3.f.f.e.4.0.0.0.2.0.0.0.0.0.0.0.ip6.arpa** para las direcciones IPv6 que deseamos que se resuelvan de forma inversa en este caso se ha añadido la dirección del servidor y del router y pulsamos en botón ***Salvar***.

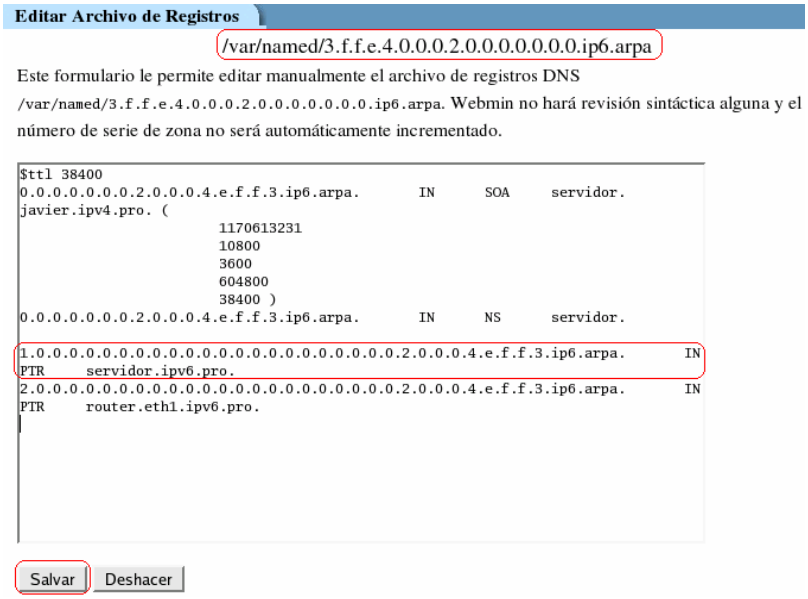


Figura 4-78: Registros añadidos para la zona inversa

5) Volvemos al índice del modulo y arrancamos el servidor DNS pulsando el botón con texto ***Arrancar Servidor de Nombres*** este botón esta localizado en la parte baja de hoja principal de configuración del Servidor DNS BIND.

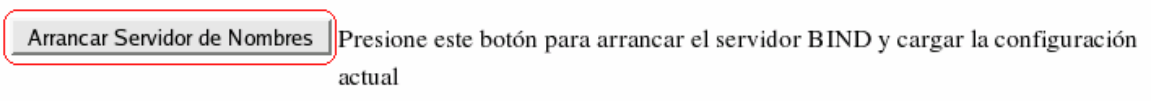


Figura 4-79: Arrancando el servidor DNS

4.4.2 Servidor DHCP

Para configurar el servidor DHCP se ha utilizado el paquete Dibbler este paquete no viene incluido en la distribución de CentOS. Para realizar la configuración de este paquete se ha realizado el siguiente procedimiento en la maquina que actúa como router:

1) Ir a la carpeta **/etc/dibbler/** y buscar el archivo **server.conf**

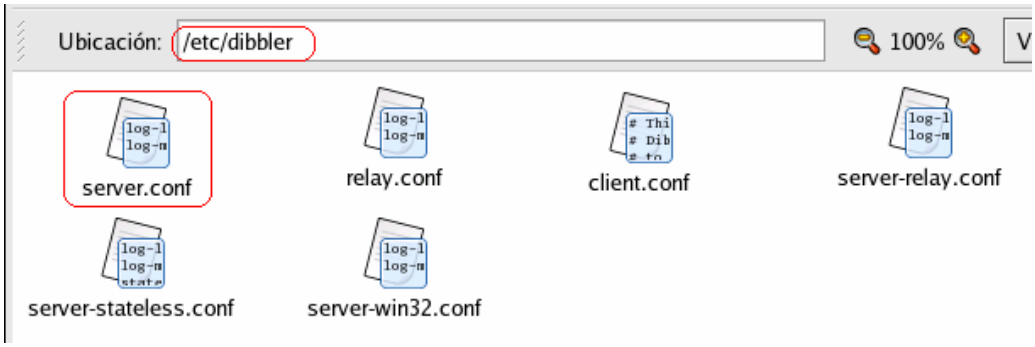


Figura 4-80: Archivo de configuración de dibbler

2) Abrir el archivo **server.conf** con cualquier editor de texto en este caso se ha utilizado el *gedit* aquí editaremos lo siguiente:

- La interface de red por la cual se realizara la asignación de direcciones IPv6 para este proyecto se ha utilizado eth1
- Los rangos de direcciones que queremos asignar en este caso lo haremos desde la 3ffe:100:100::10 hasta la 3ffe:100:100::100 La dirección del servidor DNS en este caso es 3ffe:4000:2000::1
- El dominio de red en este caso ipv6.pro y

- Finalmente *Guardamos* los cambios.

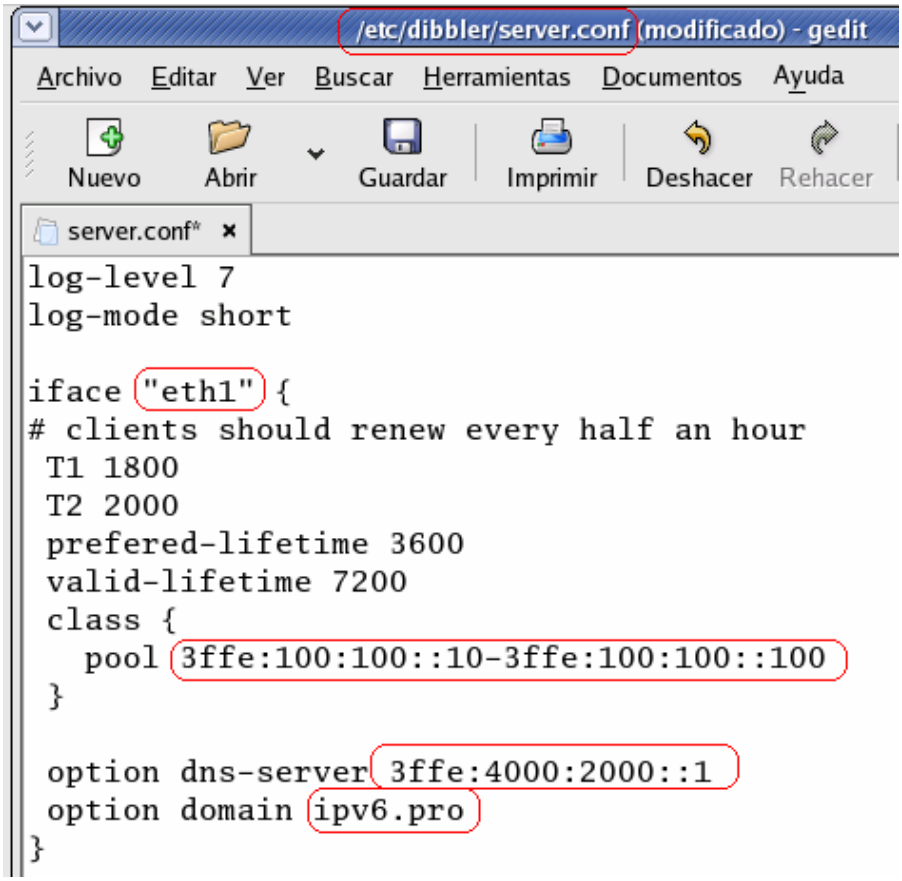


Figura 4-81: Líneas editadas en el archivo /etc/dibbler/server.conf

3) Finalmente abrimos una ventana de Terminal y arrancamos el servidor DHCP mediante el comando ***dibbler-server start***.

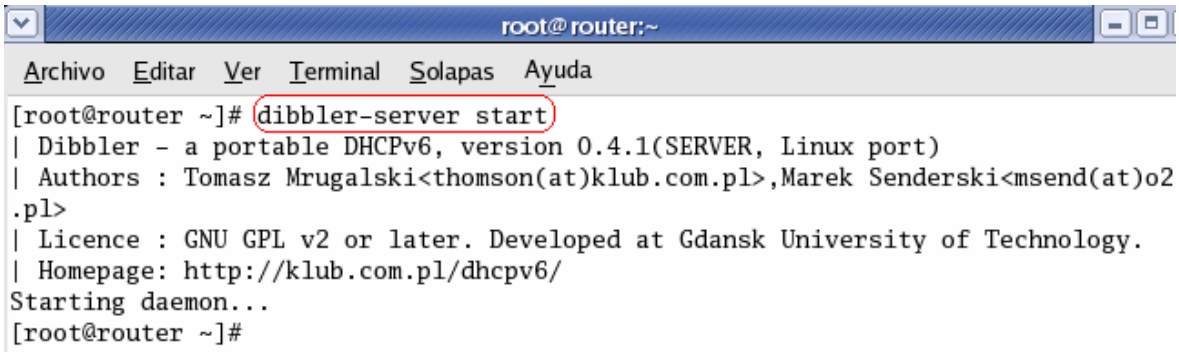


Figura 4-82: Arrancando el servidor DHCP

4.4.3 Servidor de correos

El servidor de correo utilizado es Sendmail para correo saliente y el Dovecot para correo entrante ambos incluidos en la distribución de CentOS 4.3 se los ha escogido ya que permiten abrir los puertos de escucha 25 y 110 para IPv6. Para configurar el servidor de correos se deben seguir los siguientes pasos:

4.4.3.1 Creación de usuarios

Antes de empezar a configurar sendmail se debe crear usuarios en el sistema que pertenezcan al grupo de mail. Para crear un usuario utilizando Webmin se ha realizado el siguiente procedimiento.

1) Abrimos un navegador Web cualquiera en nuestro caso se esta utilizando Mozilla Firefox y tecleamos en la barra de direcciones http://localhost:10000

2) Para acceder deberemos poner el username y pasword del administrador para nuestro proyecto el servidor tiene el username : root y el pasword : servidor

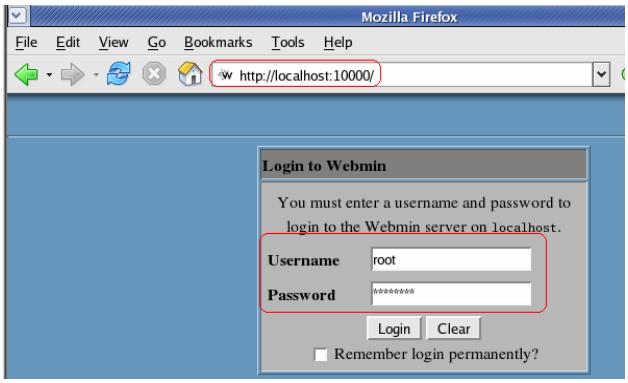


Figura 4-83: Pantalla de autentificación de Webmin

3) Damos click en la pestaña de *Sistema* y buscamos la opción de *Usuarios y Grupos* damos un click para acceder.

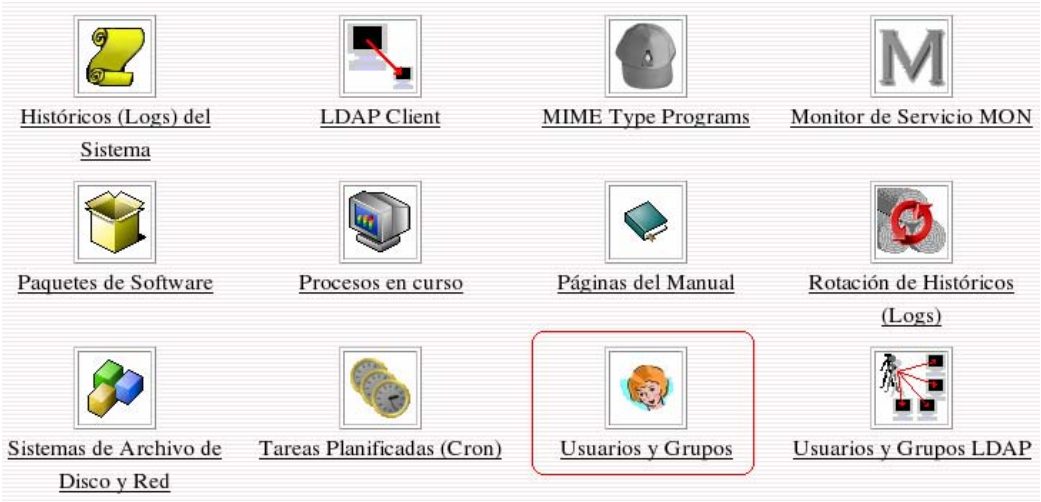
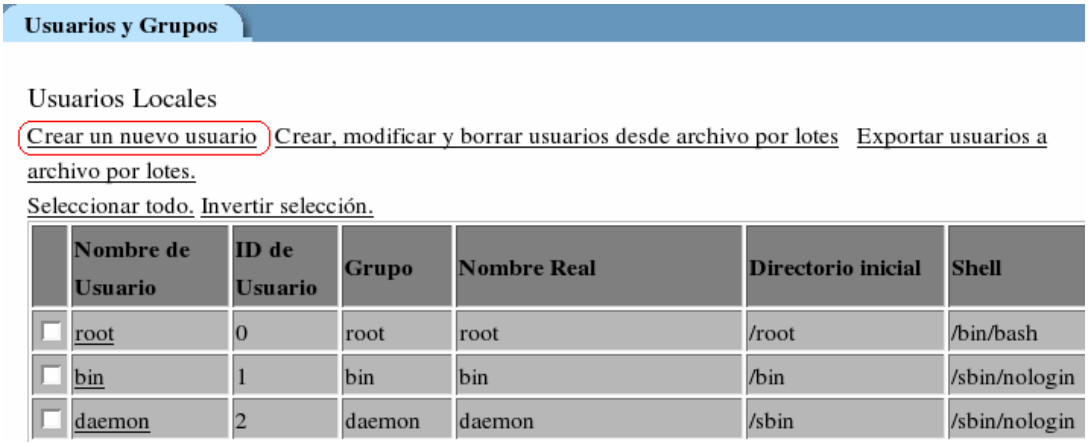


Figura 4-84: Icono de acceso a la creación de usuarios y grupos

4) Dentro de Usuarios y Grupos pulsamos en el link *Crear un nuevo usuario*.



	Nombre de Usuario	ID de Usuario	Grupo	Nombre Real	Directorio inicial	Shell
☐	root	0	root	root	/root	/bin/bash
☐	bin	1	bin	bin	/bin	/sbin/nologin
☐	daemon	2	daemon	daemon	/sbin	/sbin/nologin

Figura 4-85: Creación de un nuevo usuario

5) En los parámetros de Detalle de Usuario llenamos los siguientes campos:

- Nombre de Usuario
- Nombre Real
- Contraseña → escogemos contraseña normal y tecleamos cualquiera

Figura 4-86: Detalles de nuevo usuario

6) En Afiliación de Grupo escogemos la opción de grupo existente y ponemos ***mail*** las demás opciones las dejamos por defecto y pulsamos el botón ***Crear***.

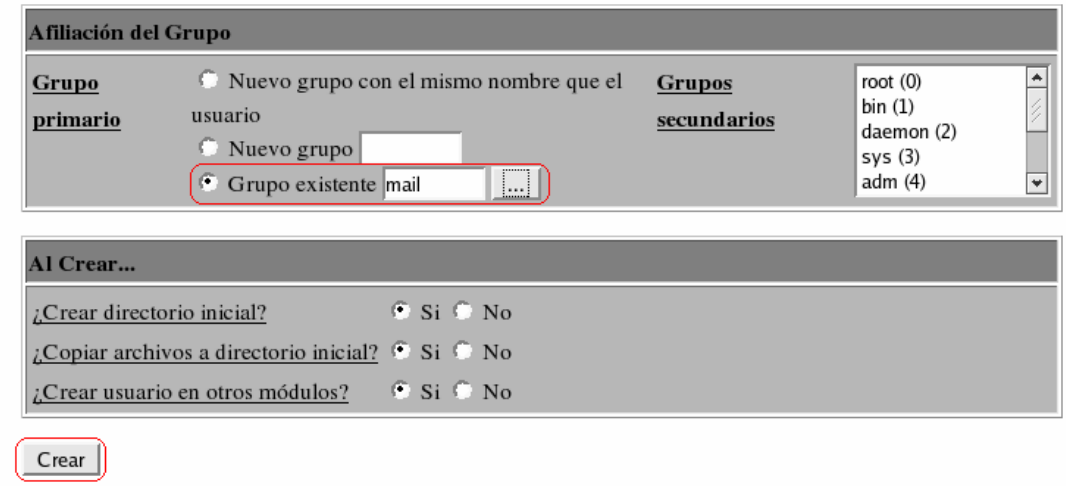


Figura 4-87: Afiliación de nuevo usuario a un grupo

4.4.3.2 Configuración de Sendmail

1) Abrimos un navegador Web cualquiera en nuestro caso se esta utilizando Mozilla Firefox y tecleamos en la barra de direcciones http://localhost:10000

2) Para acceder deberemos poner el username y pasword del administrador para nuestro proyecto el servidor tiene el username : root y el pasword : servidor

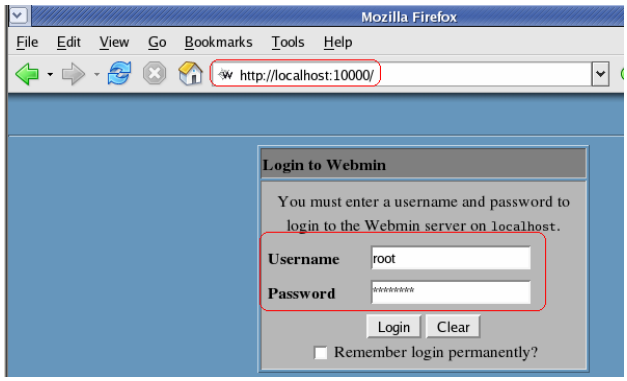


Figura 4-88: Pantalla de autentificación de Webmin

3) Pulsamos la pestaña de Servidores

4) Damos click en la opción con texto: ***Configuración de Sendmail***.

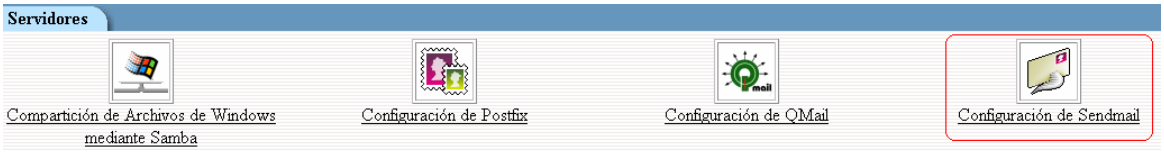


Figura 4-89: Icono de acceso a la configuración de Sendmail en Webmin

5) Dentro de la Configuración de Sendmail buscamos y damos click en la opción ***Dominios Locales (Cw)***.

Figura 4-90: Icono de acceso a la configuración de dominios locales

6) En Dominio Locales introducimos nuestro dominio que es ***ipv6.pro*** y pulsamos el botón ***Salvar***.

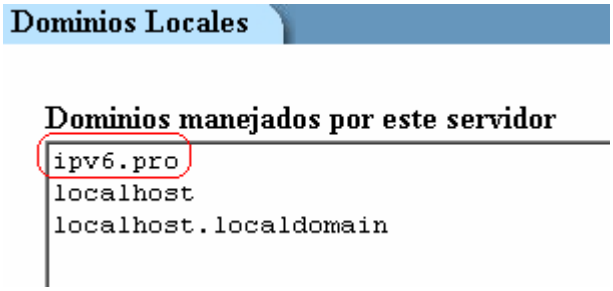


Figura 4-91: Línea de texto añadida para crear el dominio ipv6.pro

7) Volvemos al índice del modulo y nuevamente entramos en la *configuración de sendmail* y elegimos la opción de ***Usuarios Fiables (T)***.

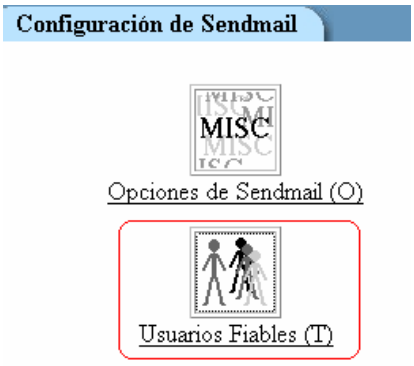


Figura 4-92: Icono de acceso para crear usuarios de correo fiables

8) Dentro de *Usuarios Fiables* añadimos los usuarios que podrán utilizar una cuenta de correo para este proyecto se ha añadido *javier* y *aylin* y pulsamos el botón ***Salvar***.

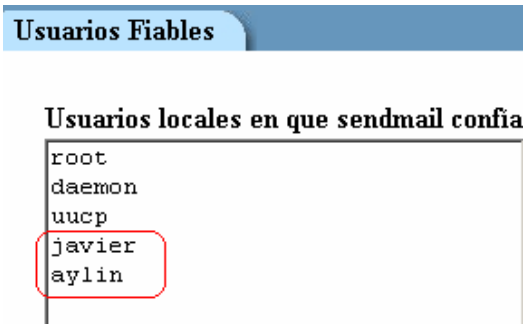


Figura 4-93: Líneas de texto añadidas para crear los usuarios

9) Regresamos al índice del modulo y entramos en la configuración de sendmail y elegimos ***Opciones de Sendmail (O)***.

Figura 4-94: Icono de acceso a las opciones de Sendmail

10) En las Opciones de Sendmail añadimos en Opciones de puerto SMTP lo siguiente: ***Name=MTA-v6, Family=inet6,Addr= 3ffe:4000:2000::1*** para que se pueda enviar correo utilizando el protocolo IPv6 y luego pulsamos el botón ***Salvar y Aplicar***.

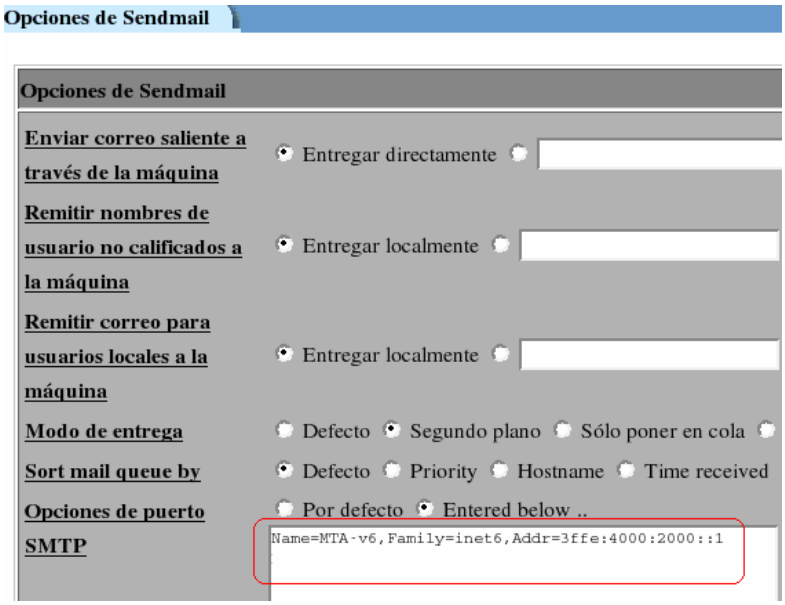


Figura 4-95: Líneas de código añadidas en las opciones de Sendmail

11) Volvemos al índice del modulo y arrancamos el servidor Sendmail pulsando el botón con texto ***Arrancar Sendmail*** este botón esta localizado en la parte baja de hoja principal de configuración del Servidor Sendmail.

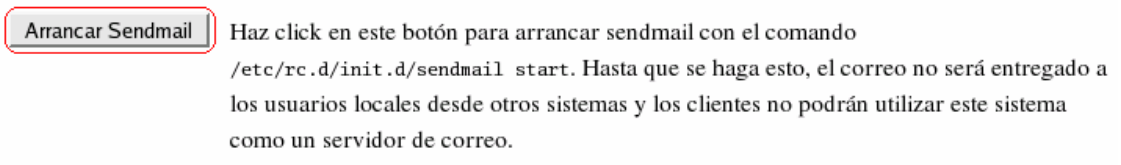


Figura 4-96: Arrancando el servidor Sendmail

4.4.3.3 Configuración de Dovecot

Para configurar Dovecot mediante Webmin se han realizado los siguientes pasos:

1) Pulsamos la pestaña de Servidores

2) Damos click en la opción con texto Dovecot : Servidor de IMAP/POP3

Figura 4-97: Icono de acceso a la configuración de Dovecot en Webmin

3) Elegimos la opción de *Red y Protocolos*.

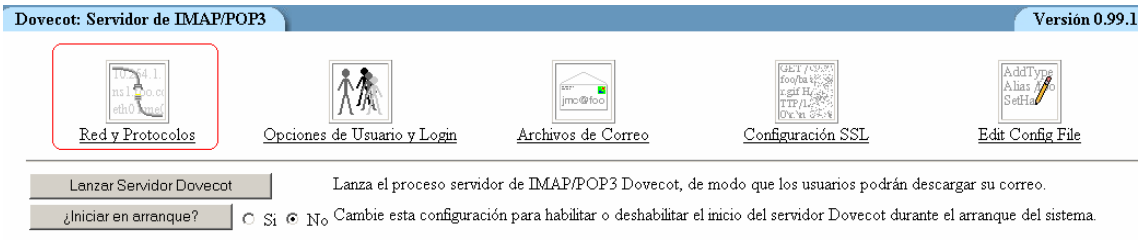


Figura 4-98: Acceso a la configuración de red de Dovecot

4) Escogemos el tipo de protocolo que manejaremos para la recepción de mail en este caso será POP3, marcamos que para todas las interfaces que se utilice para conectarse utilizando POP3 lo pueda hacer para direcciones IPv4 e IPv6 y pulsamos el botón ***Salvar*** para guardar los cambios.

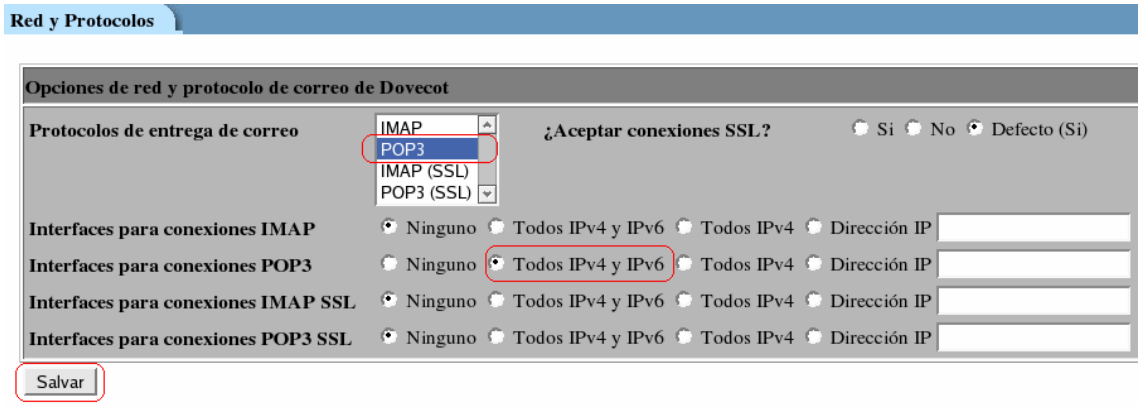


Figura 4-99: Opciones cambiadas en red y protocolos

5) Volvemos al índice del modulo de Dovecot y pulsamos sobre ***Opciones de Usuario y Login***.

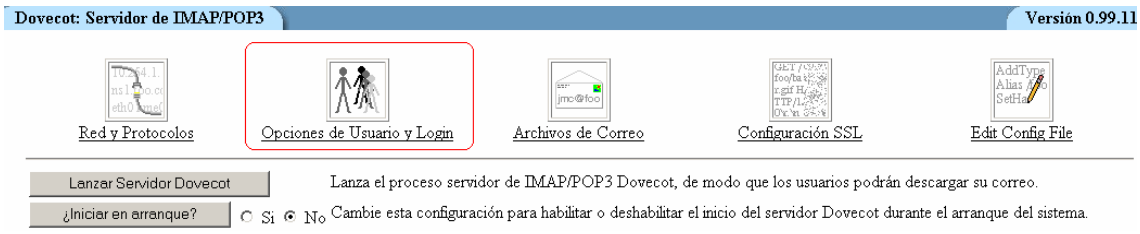


Figura 4-100: Acceso a las opciones de usuarios y login

6) Dentro de las opciones de Usuario y Login se ha configurado el método de autenticación como ***Texto Plano*** y pulsamos el botón ***Salvar***.

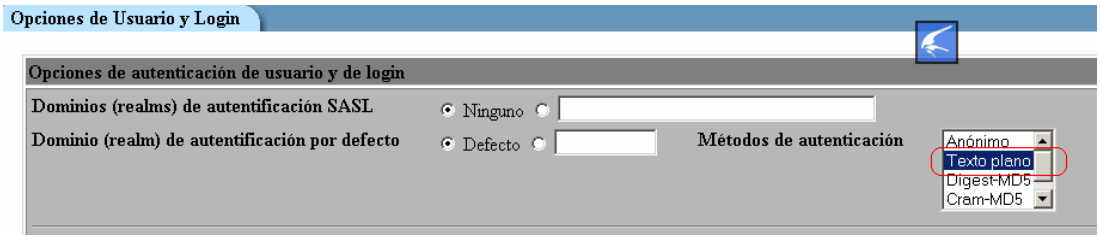


Figura 4-101: Método de autentificación escogido

7) Volvemos al índice del modulo y arrancamos el servidor Dovecot pulsando el botón con texto ***Lanzar Servidor Dovecot*** este botón esta localizado en la parte baja de hoja principal de configuración del Servidor Dovecot.

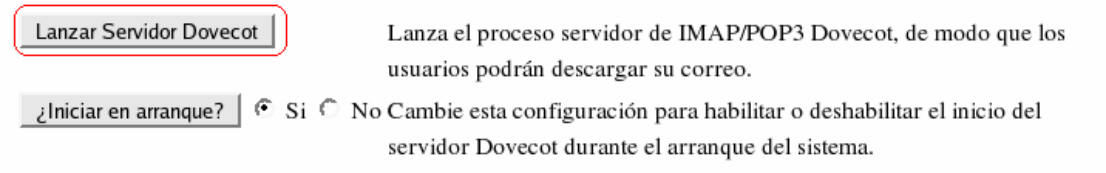


Figura 4-102: Arrancando el servidor Dovecot

4.4.4 Servidor Web

Como servidor Web se ha utilizado Apache 2.0 el cual nos permite realizar peticiones mediante direcciones IPv6 para configurarlo se ha seguido los siguientes pasos:

4.4.4.1 Creación de directorio

1) En el directorio */var/www/* creamos una carpeta con el nombre ipv6

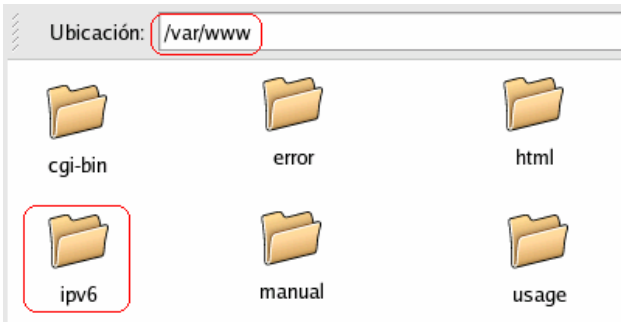


Figura 4-103: Carpeta creada en /var/www/

2) Dentro de la carpeta ipv6 creamos un archivo con nombre index.html

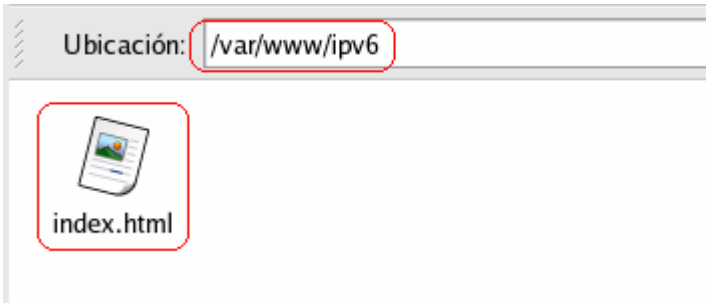


Figura 4-104: Archivo creado en /var/www/ipv6

3) Editamos el archivo index.html mediante un editor de texto, en este caso hemos puesto el texto HOLA MUNDO y guardamos los cambios.

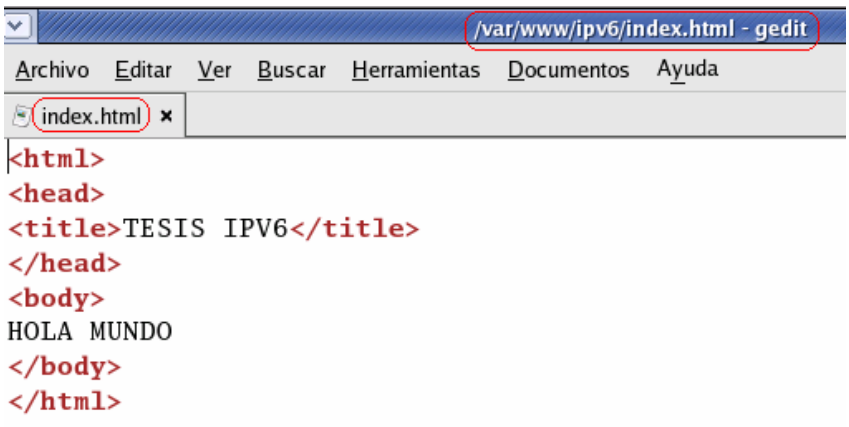


Figura 4-105: Edición del archivo index.html

4.4.4.2 Configuración de Apache

1) Abrimos un navegador Web cualquiera en nuestro caso se esta utilizando Mozilla Firefox y tecleamos en la barra de direcciones http://localhost:10000

2) Para acceder deberemos poner el username y pasword del administrador para nuestro proyecto el servidor tiene el username : root y el pasword : servidor

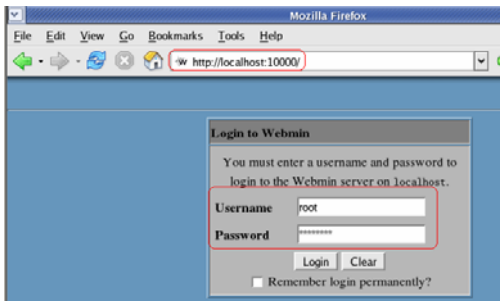


Figura 4-106: Pantalla de autentificación de Webmin

3) Pulsamos la pestaña de Servidores.

4) Damos click en la opción con texto *Servidor Web Apache*.

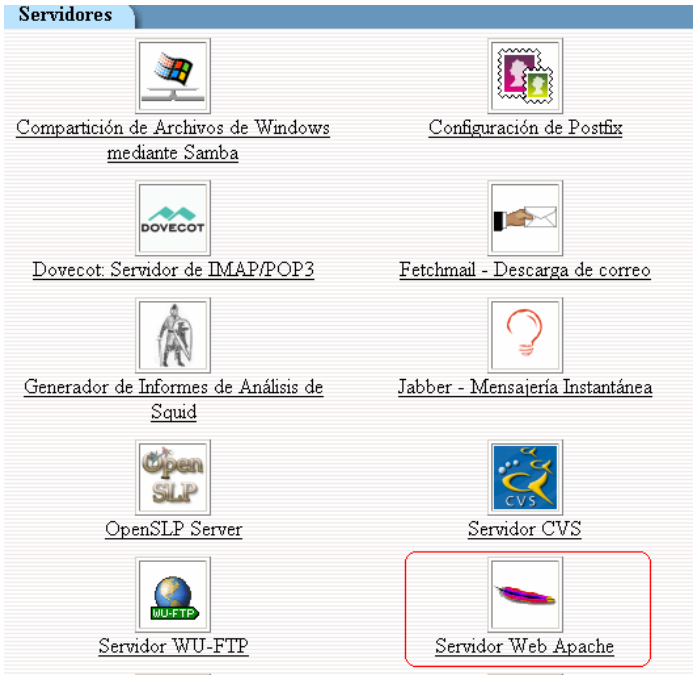


Figura: 4-107: Icono de acceso a la configuración de Apache en Webmin

5) Damos click en ***Editar Archivos de Configuración***

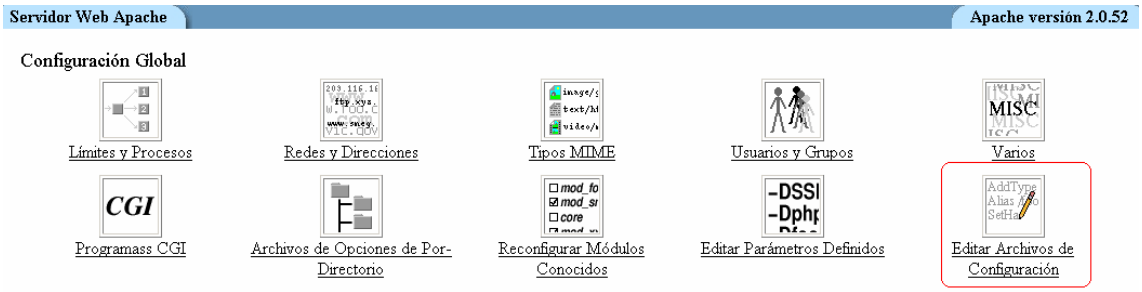


Figura 4-108: Icono de acceso a la edición de archivos de configuración

6) En *Editar Archivos de Configuración* elegimos el archivo **/etc/httpd/conf/*httpd.conf*** ,buscamos las opciones de listen y añadimos Listen [::]:80 para poder escuchar peticiones mediante IPv6 en el puerto 80.

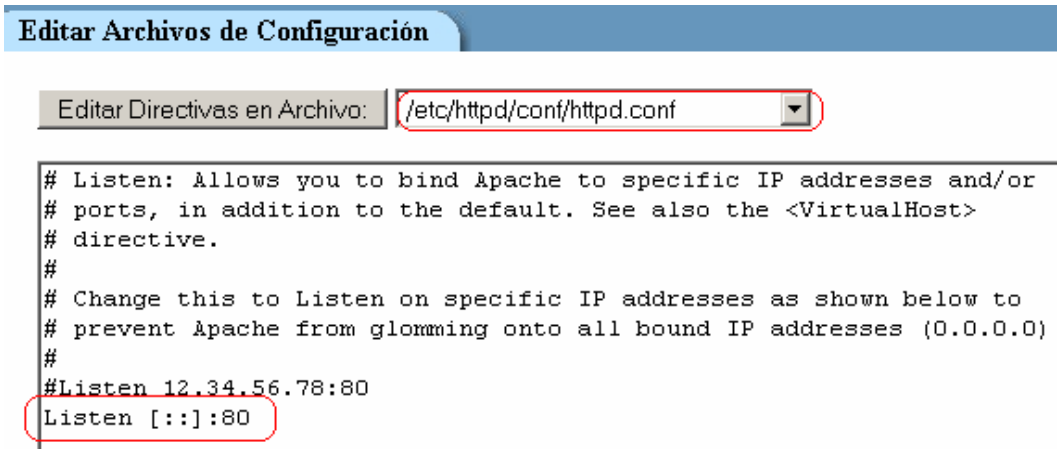


Figura 4-109: Línea de texto añadida en el archivo httpd.conf

7) Nos dirigimos hacia el final del archivo anterior **/etc/httpd/conf/*httpd.conf*** y creamos un Servidor Virtual donde especificaremos:

- Dirección especifica y el puerto de escucha (la dirección IP por la cual se accederá al contenido de los archivos puestos en la raíz para documentos)

- Raíz para documentos (aquí se especificara el lugar donde esta el contenido del sitio a mostrar)

Figura 4-110: Parámetros introducidos para crear un servidor virtual

8) El resumen de los sitios virtuales creados se lo puede observar en el índice del modulo de Apache en Servidores Virtuales.

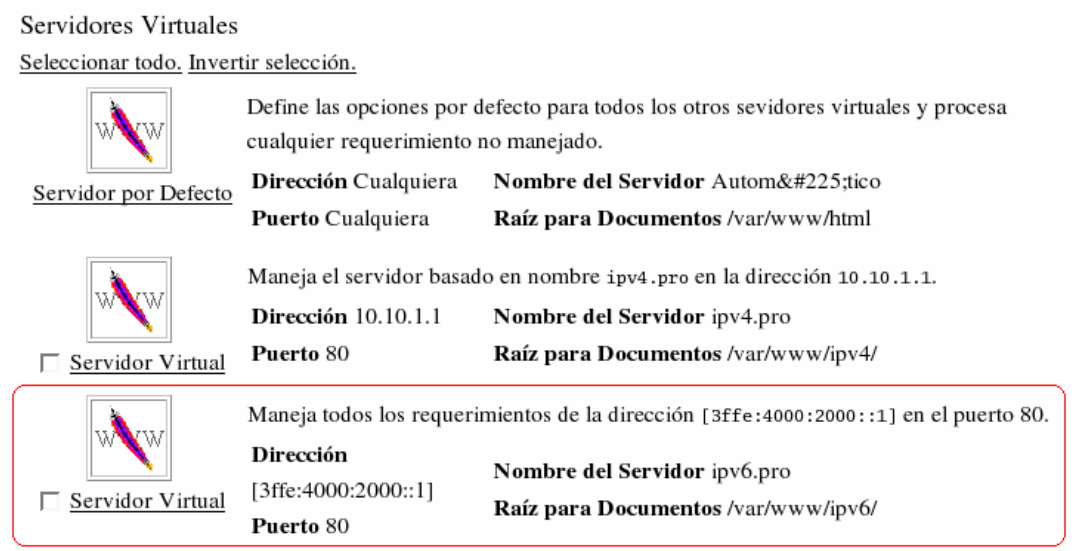


Figura 4-111: Servidores virtuales creados

9) Volvemos al índice del modulo y arrancamos el servidor Web Apache haciendo un click en el link con texto ***Arrancar Apache*** este texto esta localizado en la parte superior de la hoja principal de configuración del Servidor Web Apache.

Figura 4-112: Arrancando el servidor Web Apache

4.4.5 Servidor FTP

Para configurar el servidor FTP se ha utilizado el paquete vsftpd el cual viene incluido en la distribución de CentOS 4.3. Para realizar la configuración de este paquete se ha realizado el siguiente procedimiento:

1) El paquete *vsftpd* tiene un archivo de configuración llamado ***vsftpd.conf*** el mismo que se encuentra ubicado en ***/etc/vsftpd/vsftpd.conf***.

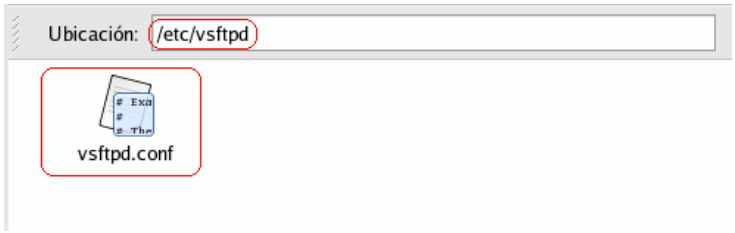


Figura 4-113: Archivo de configuración de vsftpd

2) Abrimos el vsftpd.conf con cualquier editor de texto en este caso se ha utilizado el *gedit* y añadimos las siguientes líneas de texto *listen_ipv6=YES y tcp_wrappers=YES* estas líneas permiten que el vsftpd pueda gestionar peticiones hechas desde clientes con IPv6.

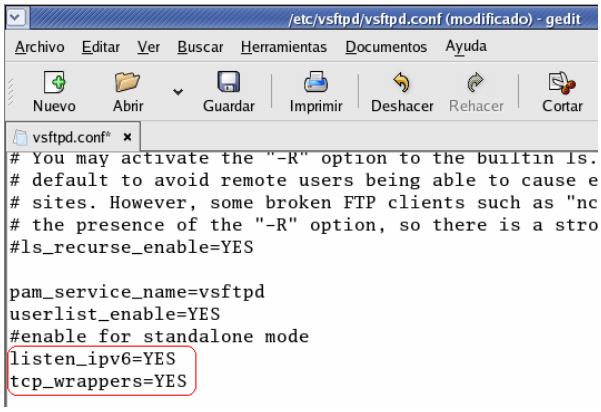


Figura 4-114: Líneas editadas en el archivo vsftpd.conf

3) Finalmente abrimos una ventana de Terminal y arrancamos el servidor FTP mediante el comando ***/etc/init.d/vsftpd start***.

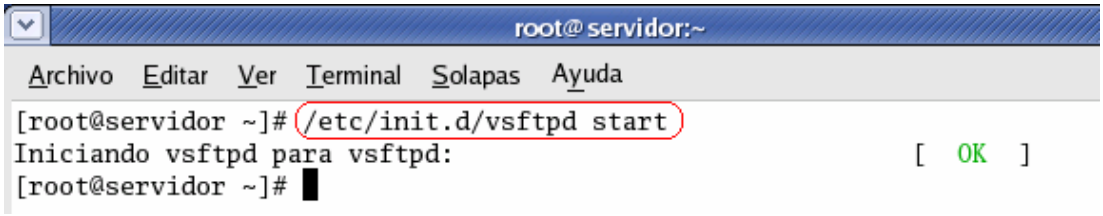


Figura 4-115: Arrancando el servidor FTP

4.4.6 Servidor SSH

Para configurar el servidor de SSH mediante Webmin se han realizado los siguientes pasos:

1) Abrimos un navegador Web cualquiera en nuestro caso se esta utilizando Mozilla Firefox y tecleamos en la barra de direcciones http://localhost:10000

2) Para acceder deberemos poner el username y pasword del administrador para nuestro proyecto el servidor tiene el username : root y el pasword : servidor

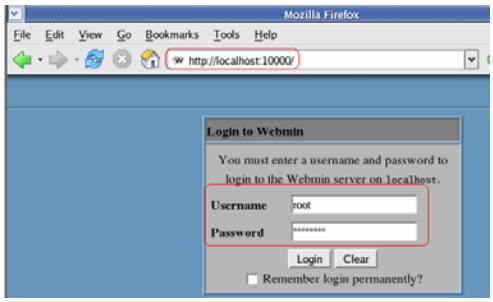


Figura 4-116: Pantalla de autentificación de Webmin

3) Pulsamos la pestaña de Servidores.

4) Damos click en la opción con texto *Servidor SSH*.

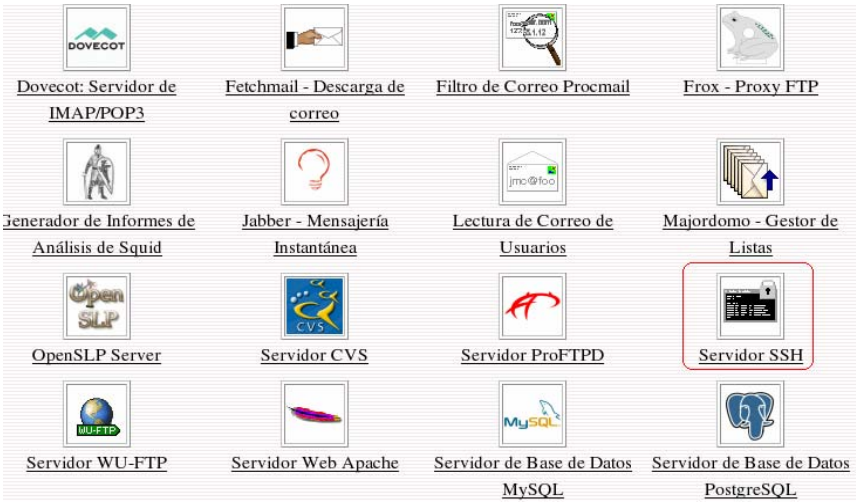


Figura: 4-117: Icono de acceso a la configuración de SSH en Webmin

5) Damos click en ***Editar Archivos de Configuración***.

Figura: 4-118: Icono de acceso a la configuración de archivos

6) Editamos el archivo ***sshd-config*** dando click en editar ***/etc/ssh/sshd-config*** en este archivo se ha descomentado las siguientes líneas de texto ***Port 22*** y ***ListenAddress ::*** para que pueda soportar peticiones IPv6. Finalmente pulsamos el botón Salvar para guardar los cambios.

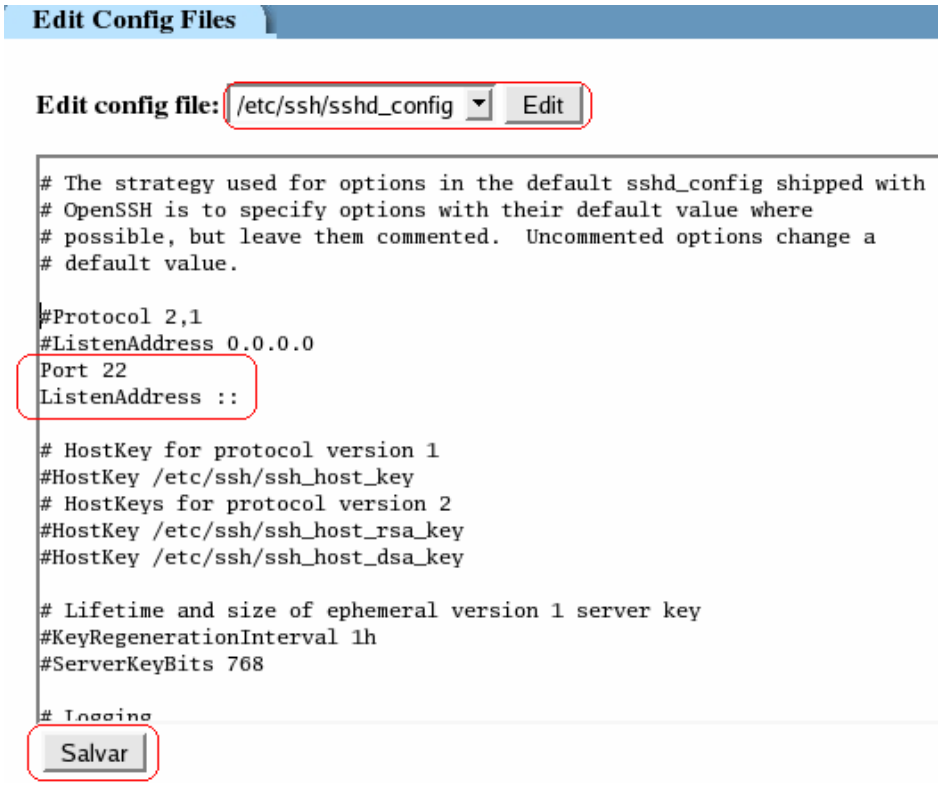


Figura: 4-119: Líneas editadas en el archivo sshd_config

7) Volvemos al índice del modulo y arrancamos el servidor SSH pulsando el botón con texto ***Arrancar Servidor*** este botón esta localizado en la parte baja de hoja principal de configuración del Servidor Dovecot.

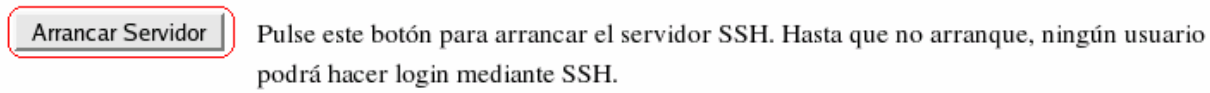


Figura 4-120: Arrancando el servidor SSH

4.5 Configuración de Clientes

Los paquetes de software clientes que se han utilizado tanto en el servidor como en las computadoras clientes para realizar las pruebas de los servicios que presta cada servidor levantado son los que se detalla en la Tabla 4-3:

Tabla 4-3: Software utilizado para los clientes

Cliente	Paquete	Plataforma soportada
Web	Mozilla Firefox	Windows y Linux
Correo	Evolution	Windows y Linux
DHCP	Dibbler	Windows y Linux

4.5.1 Cliente Web

Como cliente Web se ha utilizado Mozilla Firefox el cual soporta el protocolo IPv6, para que el programa funcione adecuadamente se debe verificar la configuración siguiente:

1) Abrir el programa Mozilla Firefox.

2) Verificamos que las configuraciones de la conexión este en la opción de forma directa a Internet para ellos vamos al menú de *Herramientas* y seleccionamos *Opciones*.

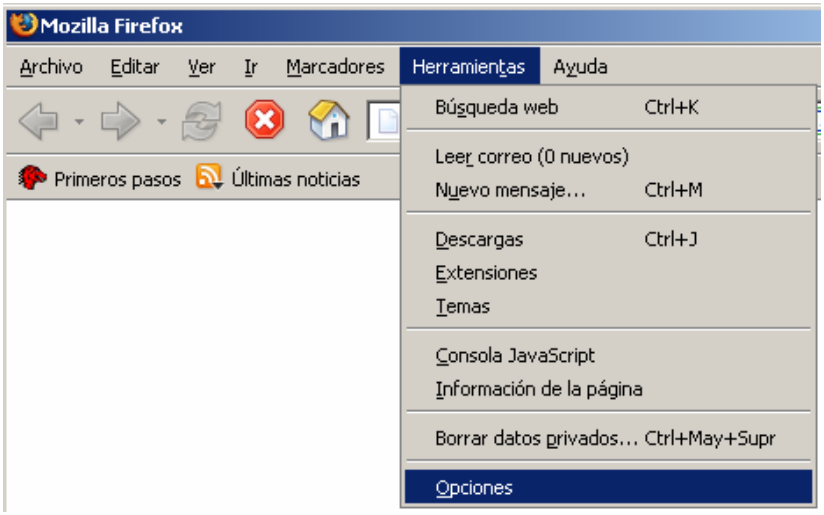


Figura 4-121: Menú de herramientas de Mozilla Firefox

3) En la ventana con titulo de Opciones pulsamos el botón Configuración de conexión.

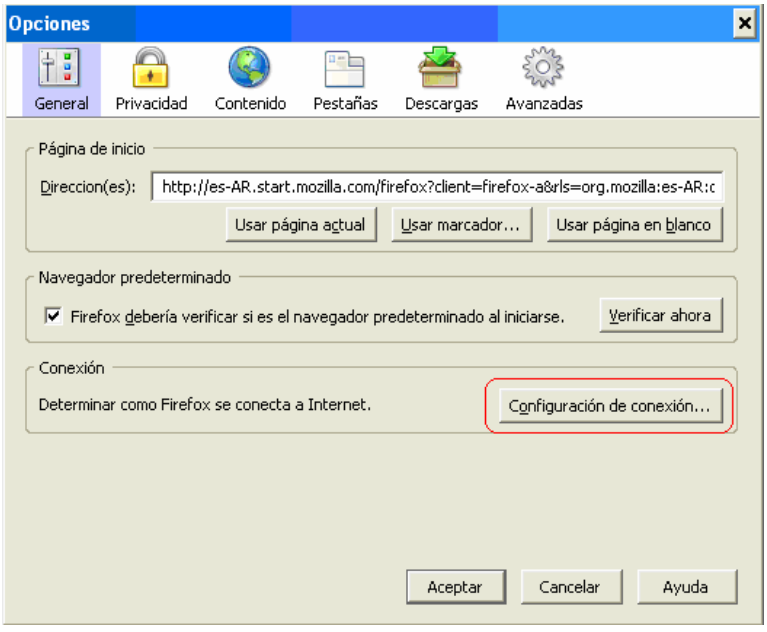


Figura 4-122: Configuración de conexión en Mozilla Firefox

4) En la ventana de Configuración de la conexión seleccionamos ***configuración directa a Internet***.

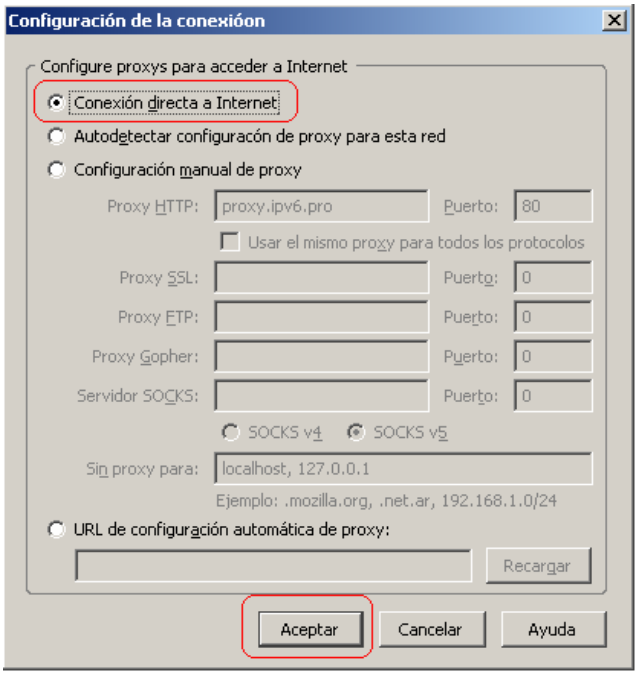


Figura 4-123: Ventana de configuración de la conexión

5) En la barra de dirección tecleamos la dirección IPv6 del servidor esta debe estar entre corchetes (Figura 4-124).

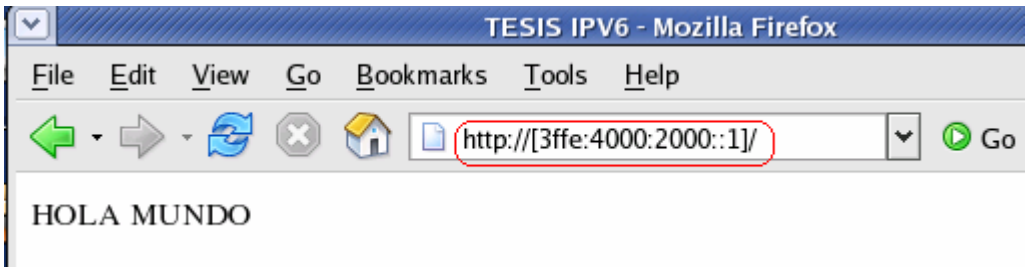


Figura 4-124: Página Web abierta mediante IPv6

4.5.2 Cliente de Correo

Como cliente de correo se ha utilizado ***Evolution*** el cual viene incluido en la distribución de Fedora Core 5 en este caso para realizar las pruebas hacia el servidor de correo se ha configurado una cuenta de correo en la PC cliente que tiene sistema operativo Fedora Core 5. Para configurar una cuenta de correo en Evolution se han realizado los siguientes pasos:

1) Ejecutar Evolution accediendo por el menú Aplicaciones→ Internet → Correo electrónico

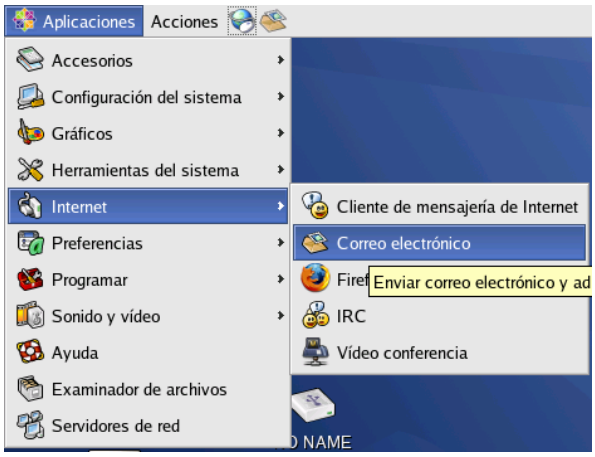


Figura 4-125: Ejecución de correo electrónico Evolution

2) Ir al menú de Herramientas → Configuración

Figura 4-126: Menú de configuración de Evolution

3) En las preferencias de Evolution escogemos del lado izquierdo ***Cuentas de correo*** y pulsamos el botón del lado derecho que dice ***Añadir***.

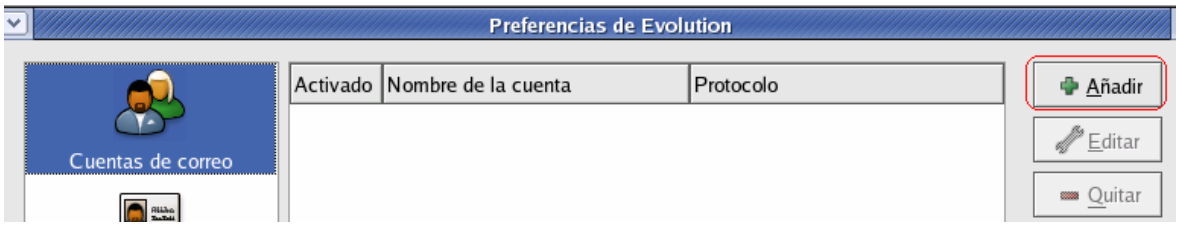


Figura 4-127: Creación de una cuenta de correo en Evolution

4) Se abrirá una pantalla en la cual nos dice que sigamos las instrucciones del asistente de configuración damos click en el botón Adelante y nos presenta una ventana en la cual llenaremos el nombre que mostrara la cuenta y la dirección de correo.

Figura 4-128: Información de identidad requerida por Evolution

5) Elegimos el tipo de servidor en este caso para nuestro proyecto es de tipo *POP*.

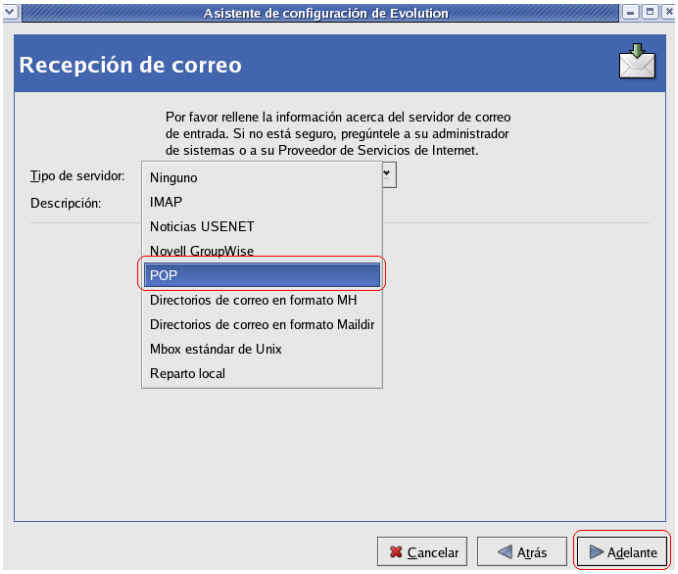


Figura 4-129: Tipo de servidor utilizado para la recepción de correo

6) Llenamos cual es nuestro servidor POP y el tipo de autenticación requerida por el servidor para nuestro proyecto el servidor POP es ***pop3.ipv6.pro*** y el tipo de autenticación es por contraseña.

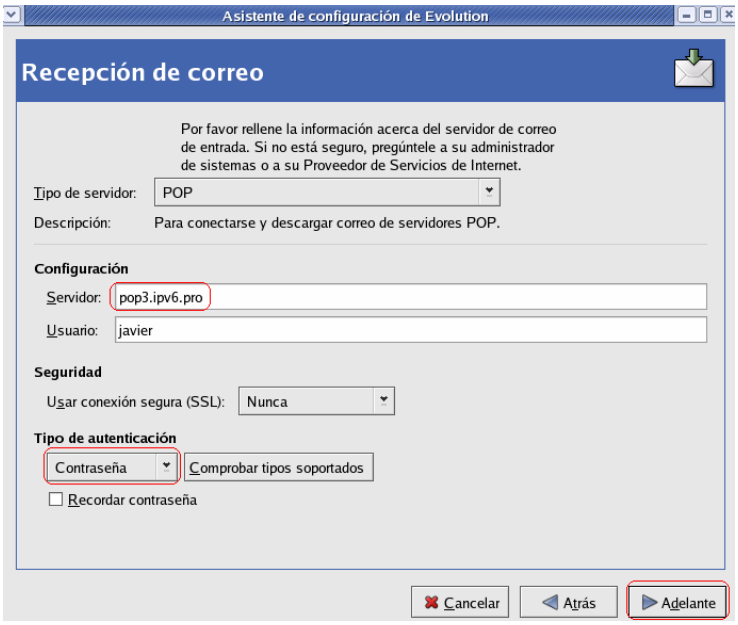


Figura 4-130: Parámetros de configuración del servidor de correo entrante

7) Escogemos el tipo de servidor de correo saliente en nuestro caso es tipo *SMTP* y ponemos el nombre ó la dirección del servidor en nuestro caso este dato será ***smtp.ipv6.pro*** los demás datos los dejamos en blanco puesto que no utilizamos autenticación.

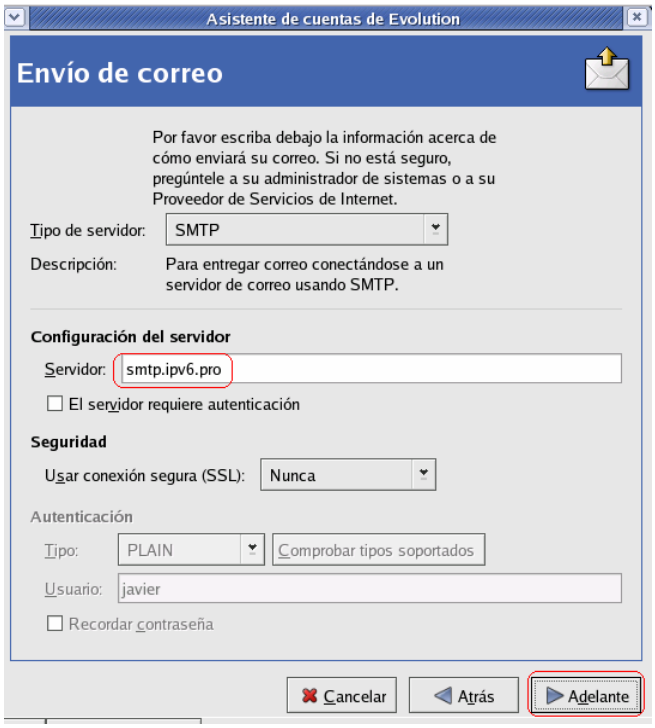


Figura 4-131: Parámetros editados en configuración del servidor SMTP

4.5.3 Cliente DHCPv6

Como cliente DHCPv6 se ha utilizado Dibbler el cual se lo ha instalado y configurado en Windows XP SP2 para realizar las pruebas hacia el servidor DHCP. Para la configuración del cliente utilizamos el archivo ***client.conf*** en el cual configuramos:

1) Damos click en Inicio Programas → Dibbler → Client Edit config file

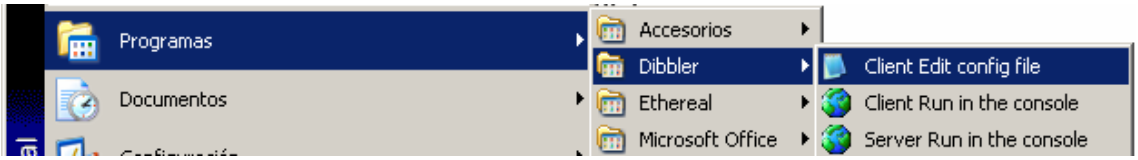


Figura 4-132: Acceso directo por el menú de inicio al archivo client.conf

2) En el archivo *client.conf* ponemos el nombre de la interface de red que realizara la petición de una dirección IPv6 al servidor DHCP en nuestro caso es ***Conexión de área local*** y ***Guardamos*** los cambios.

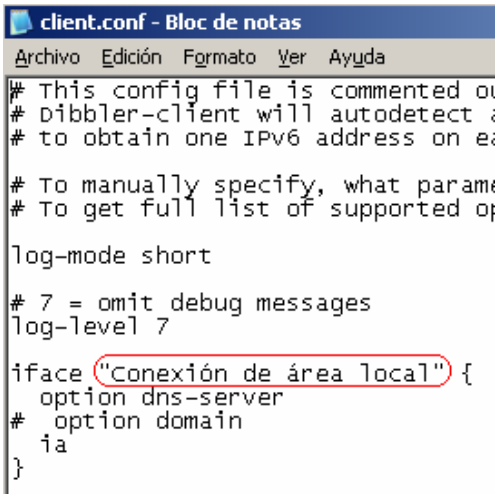


Figura: 4-133: Líneas editadas en el archivo client.conf

3) Para ejecutar el cliente de DHCP vamos al menú de Inicio → Programas → Dibbler → Client Run in the console.

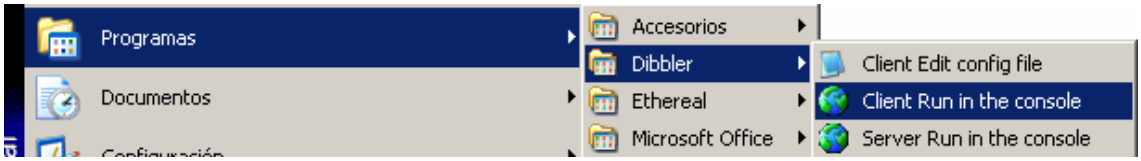


Figura: 4-134: Inicialización de la consola de Dibbler

4) Si tenemos Windows XP SP2 nos saldrá un mensaje como el mostrado en la Figura 4-135 en este caso tendremos que pulsar la opción desbloquear para que Dibbler pueda realizar peticiones de direcciones IPv6 al servidor.

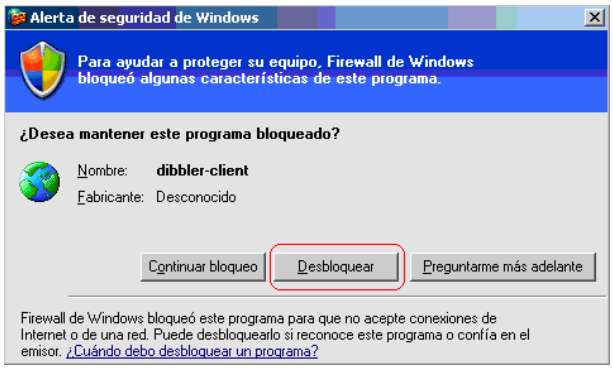


Figura 4-135: Mensaje de firewall de Windows XP SP2

5) Luego en la consola del cliente podemos observar una notificación en la que indica que ha sido asignada una dirección a nuestra inteface de red.

```
opts: 1 3 2 23
13:10:17 Notice    Address 3ffe:4000:2000::5a added to Conexi%n de βrea local/
interface.
13:10:17 Notice    Setting up DNS server 3ffe:4000:2000::1 on interface Conexi
 de βrea local/4.
13:10:17 Notice    Sleeping for 1 second(s).
```

Figura 4-136: Mensaje en la consola de cliente de Dibbler

6) Para verificar que el servidor nos asigno a la interface de red la dirección que nos indica en la consola del cliente DHCP abrimos una ventana de comandos e introducimos el comando ***ipconfig***.

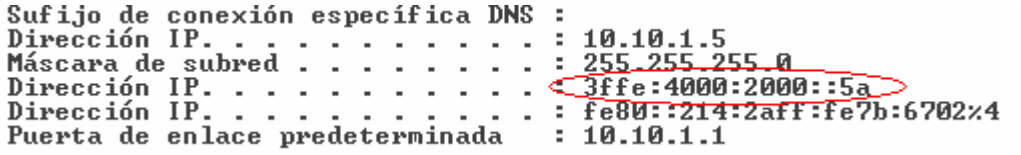


Figura 4-137: Verificación de la IP asignada mediante el comando ipconfig

4.6 Configuración de QoS

Para que Linux pueda realizar control de tráfico se debe verificar que en el kernel estén activadas todas las opciones de ***Netfilter Configuration y QoS and/or fair queueing.***

4.6.1 Verificación de la configuración del kernel para el manejo de QoS

Los pasos que se deben seguir para verificar que la configuración del kernel sea la correcta se describe a continuación:

1) Abrimos una ventana de Terminal y nos dirigimos a ***/usr/src/kernels/2.6.9-34.EL-smp-i686*** y ejecutamos el comando ***make xconfig***

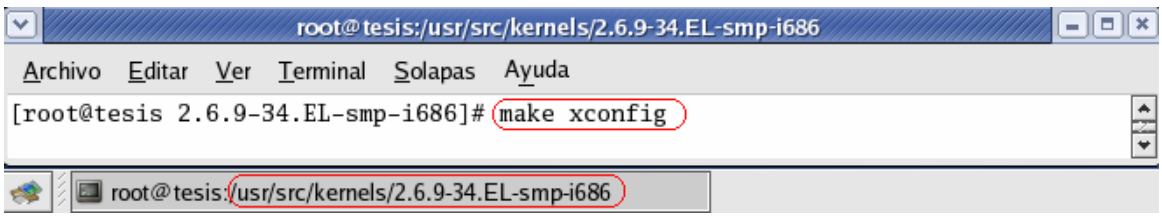


Figura. 4-138: Ejecución del comando xconfig en una ventana de Terminal

2) Hacer doble click sobre la opción ***Device Drivers*** para abrir el árbol.

3) Dentro de ***Device Drivers*** buscar la opción de ***Networking support*** y la señalamos con un click para que se nos despliegue las opciones de esta en el lado derecho.

4) En las opciones de ***Networking support*** del lado derecho damos un doble click en la que dice ***Networking options***

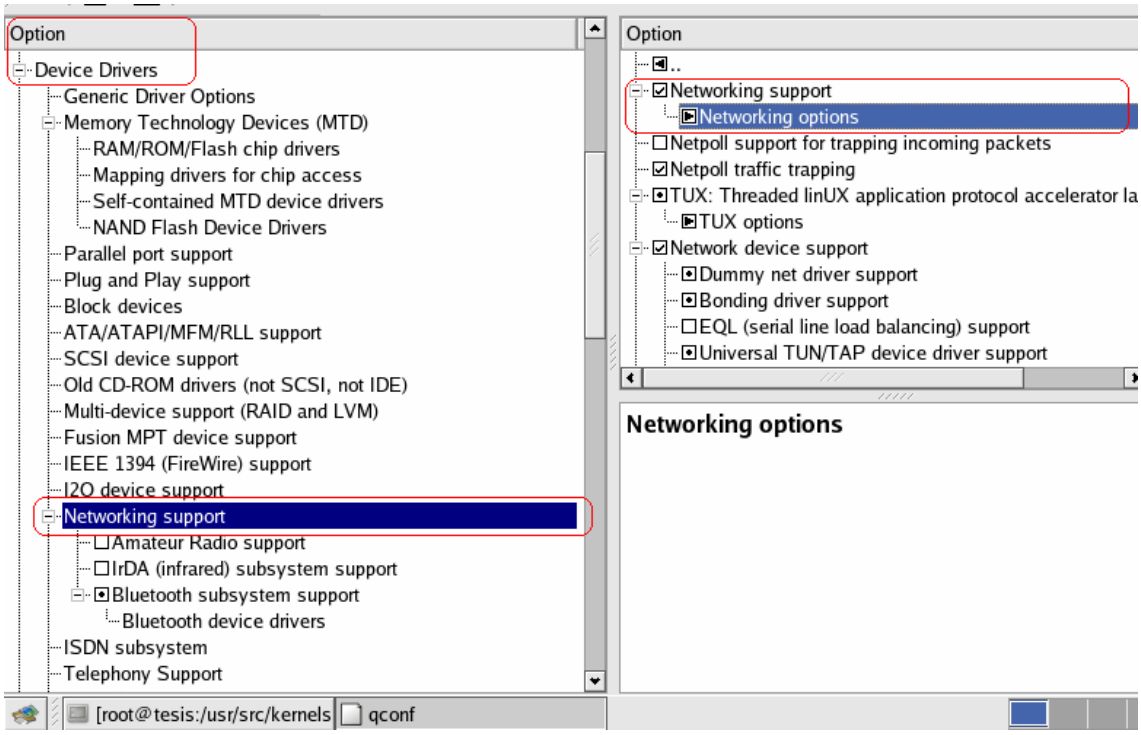


Figura 4-139: Opciones de configuración de Networking

5) Dentro de las opciones de ***Networking options*** damos doble click en ***Network packet filtering (replaces ipchains)***.

6) Una vez desplegadas las opciones de Network packet filtering (replace ipchains) hacemos doble click sobre ***IPv6: Netfilter Configuration***.

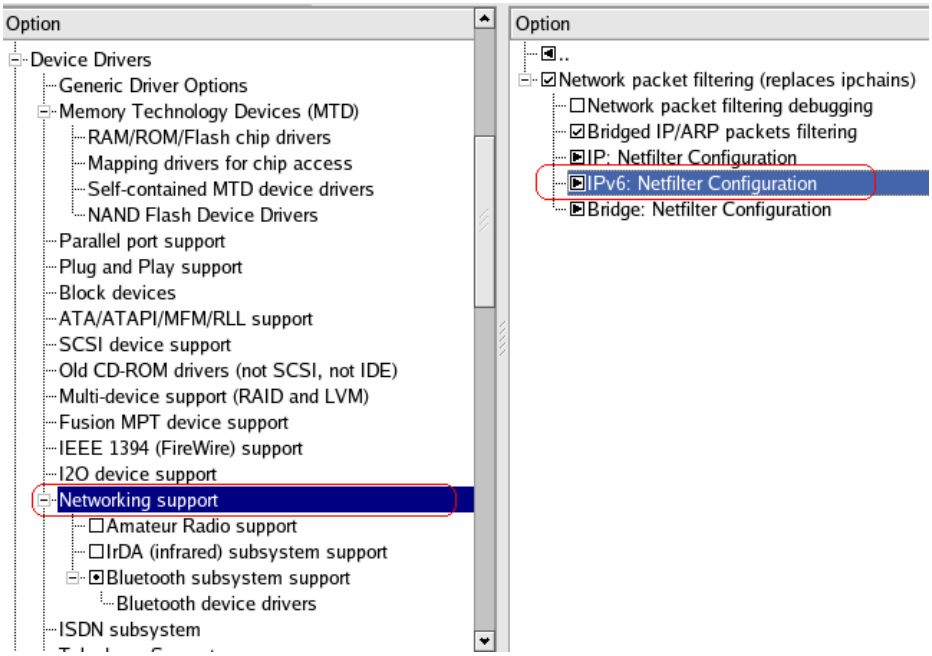


Figura. 4-140: Opciones de configuración de IPv6 Netfilter

7) En las opciones que se despliegan en ***IPv6: Netfilter Configuration*** deben estar señaladas todas las que están dentro de ***IPv6 tables support***.

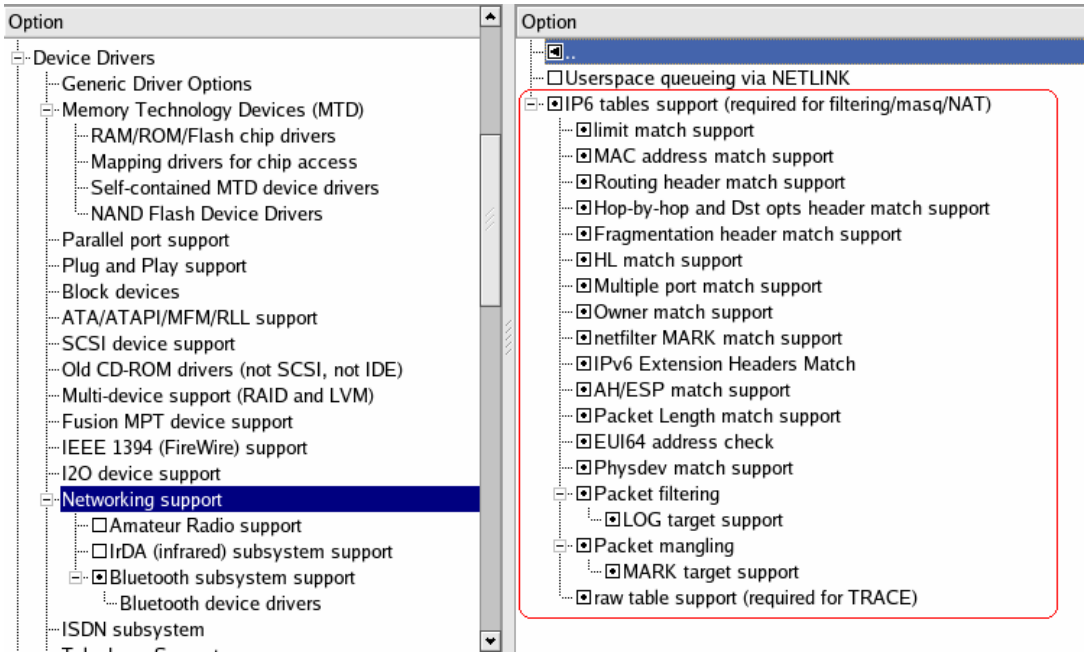


Figura 4-141: Módulos marcados en la configuración de IPv6 Netfilter

8) Nuevamente damos doble click en el lado izquierdo en la opción de ***Networking support*** y escogemos en el lado derecho ***QoS and/or fair queueing***.

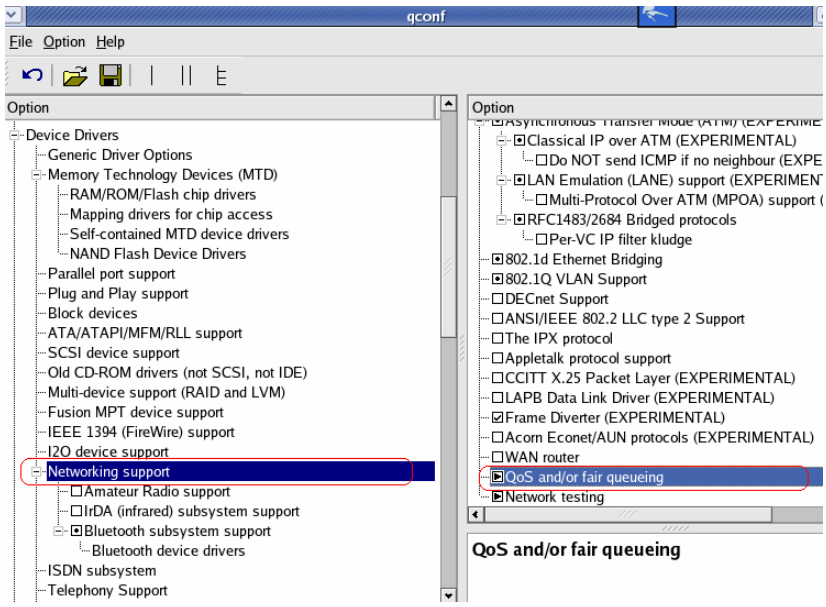


Figura 4-142: Opciones de configuración de QoS

9) Una vez desplegada la opción de ***QoS and/or fair queueing*** verificamos que estén marcadas todas las opciones.

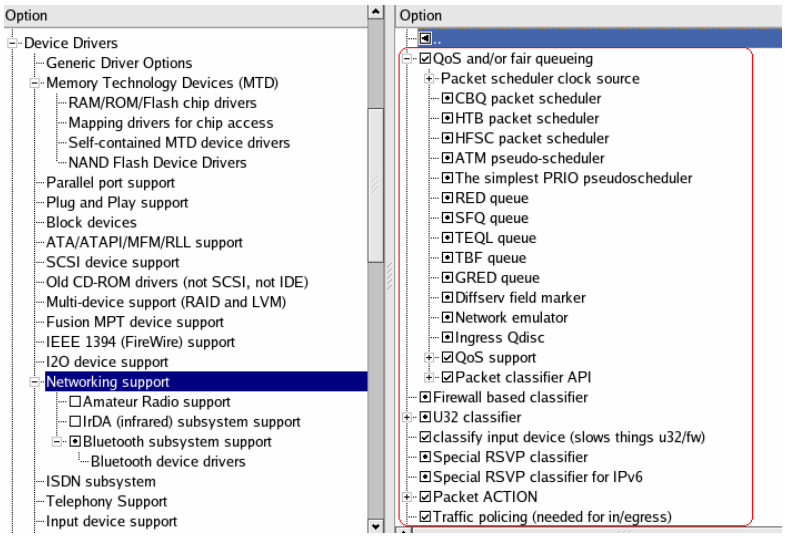


Figura 4-143: Módulos marcados en la configuración de QoS

10) Finalmente salimos sin guardar ningún cambio pues la versión de CentOS 4.3 ya trae todas las opciones de control de tráfico activadas, si se realizara alguna modificación extra se deberá compilar el kernel e iniciar el sistema operativo con la nueva versión del kernel personalizado.

4.7 Implementación de QoS

Para la implementación de QoS en linux se ha seguido el diseño del árbol de clases de la Figura. 4-144 el cual nos ha permitido implementar las políticas de QoS.

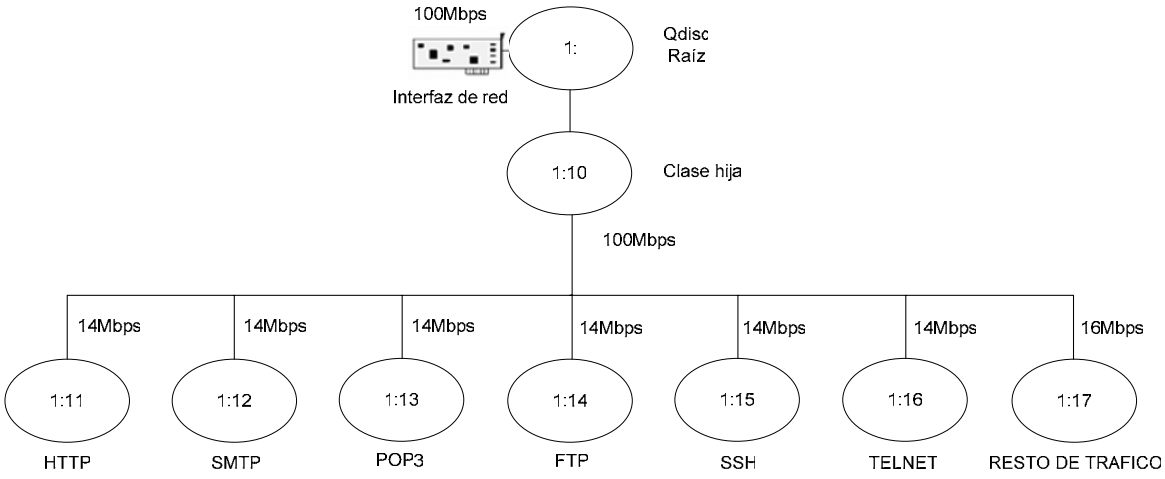


Figura 4-144: Diseño del árbol de QoS para los diferentes servicios

4.7.1 Creación de la raíz del árbol

Para la creación del árbol de bandas del diseño utilizaremos un archivo de script llamado ***filter.sh*** el cual estará ubicado en la raíz del sistema y al cual iremos añadiendo las líneas de comando correspondientes para crear las qdisc, clases y filtros todo esto se lo hará utilizando **tc** para configurar el control de trafico en el kernel. El procedimiento realizado para inicializar el archivo de script, crear la qdisc raíz y una clase hija se describe a continuación:

1) Definir la variable de la interface de red por la cual ingresara el tráfico de red a controlar en este caso es ***DEV="eth0"***

2) Definir la variable de la dirección de red que se utilizaran para controlar el tráfico en este caso es ***DIR="3ffe:100:100::0/64"***

3) Borrar la qdisc tipo root que vienen por default en la interface de red definida anteriormente mediante el comando ***tc qdisc del dev $DEV root***

4) Introducir la línea de comandos ***tc qdisc add dev $DEV root handle 1: htb default 17*** para crear la qdisc raíz con una disciplina de colas tipo HTB.

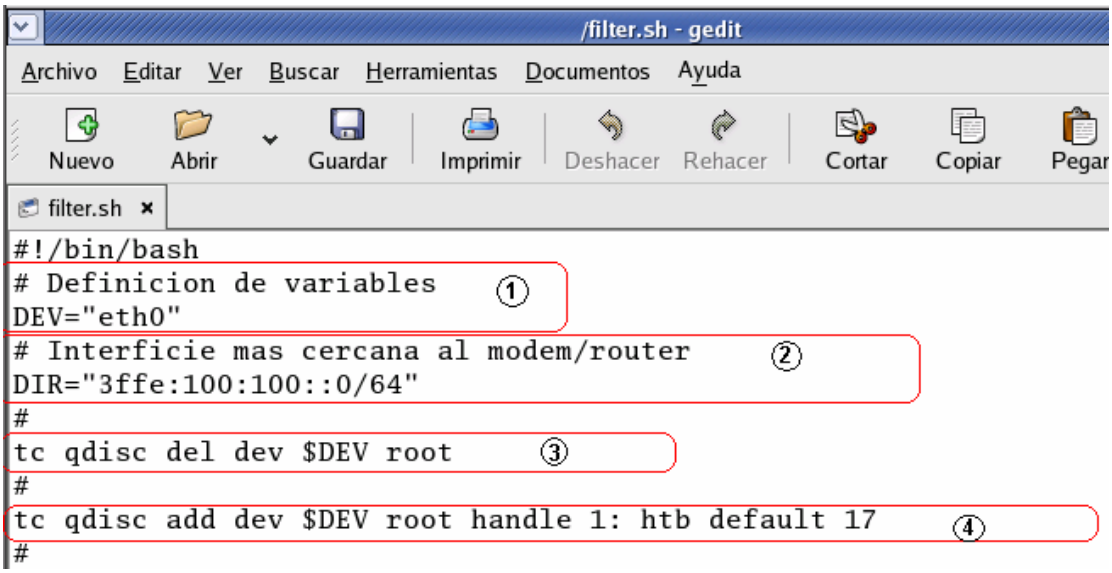


Figura 4-145: Creación de la qdisc raíz

5) Luego de creada la raíz se seguirá creando el resto del árbol en nuestro caso crearemos la clase hija 1:10 introduciendo en el archivo *filter.sh* la línea de código mostrada en la Figura 4-146.

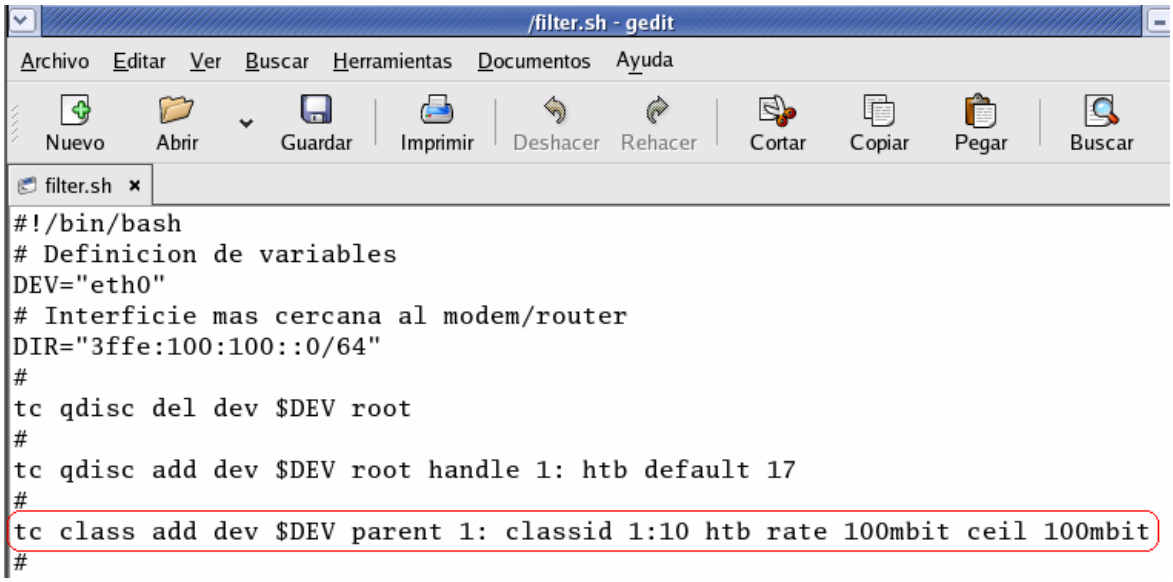


Figura 4-146: Creación de la clase hija 1:10

4.7.2 Creación de las clases

Según nuestro diseño se debe crear siete clases para los siguientes servicios:

1. Servidor Apache (puerto 80)
2. Servidor de correo SMTP (puerto 25)
3. Servidor de correo POP3 (puerto 110)
4. Servidor FTP (puerto 21)
5. Servidor SSH (puerto 22)
6. Servidor Telnet (puertos 20)
7. Resto de tráfico

Para la creación de las clases se ha utilizado el comando ***tc*** y se ha realizado un archivo de script llamado ***filter.sh*** en donde digitaremos todos los comandos necesarios para la creación de las clases esto se lo hace así dado que la introducción de estas líneas en una ventana de comandos seria muy tedioso y tendríamos que introducirlas cada vez que se reinicie el sistema operativo.

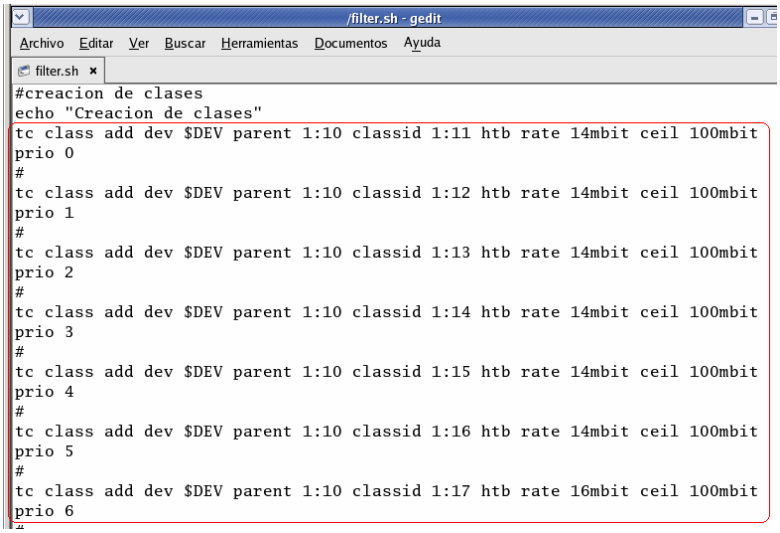


Figura 4-147: Líneas de código digitados en el archivo de script filter.sh

4.7.3 Adicionamiento de Qdisk

Luego de creada las clases crearemos las qdisc para cada una de las clases, igualmente se ha utilizado el comando **tc** igualmente se ha realizado la introducción de las siguientes líneas en el archivo de script ***filter.sh***.

filter.sh

```
#Adicionando qdisc a las clases
echo "Añadiendo qdisc a las clases"
tc qdisc add dev $DEV parent 1:11 handle 11: sfq perturb 10
#
tc qdisc add dev $DEV parent 1:12 handle 12: sfq perturb 10
#
tc qdisc add dev $DEV parent 1:13 handle 13: sfq perturb 10
#
tc qdisc add dev $DEV parent 1:14 handle 14: sfq perturb 10
#
tc qdisc add dev $DEV parent 1:15 handle 15: sfq perturb 10
#
tc qdisc add dev $DEV parent 1:16 handle 16: sfq perturb 10
#
tc qdisc add dev $DEV parent 1:17 handle 17: sfq perturb 10
#
```

Figura 4-148: Comandos digitados para añadir qdisc a las clases

4.7.4 Creación de filtros

Una vez creado el árbol de preferencias se debe configurar los filtros dentro de la qdisc raíz para que se envié los paquetes a las clases correspondientes. Para este proyecto se ha utilizado filtros en base a la dirección y puerto para esto se ha introducido las siguientes líneas en el archivo de script ***filter.sh***.

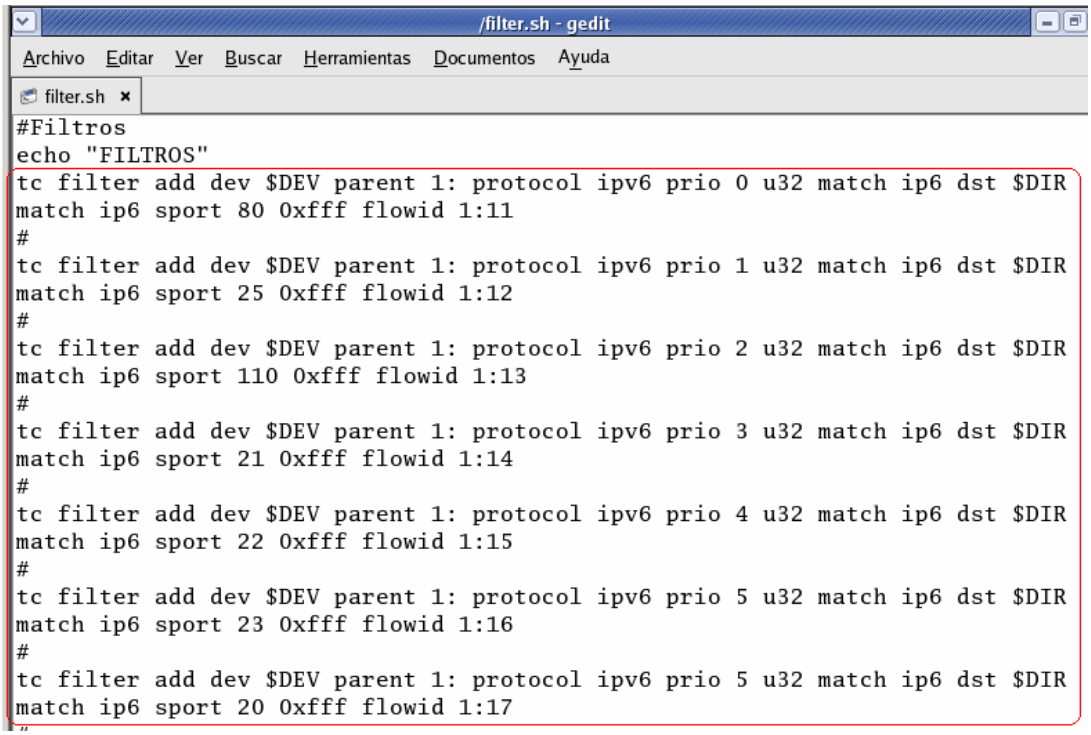


Figura 4-149: Comandos utilizados para crear filtros

4.7.5 Ejecución del archivo filter.sh

Para correr el archivo de configuración de QoS llamado ***filter.sh*** se debe acceder al directorio donde lo hemos creado y ejecutar el comando **sh** *nombre_del_archivo.* En nuestro caso el archivo esta creado en la raíz por lo tanto se accedió a el tecleando el comando ***cd ..*** y luego el comando ***sh filter.sh*** para ejecutarlo.

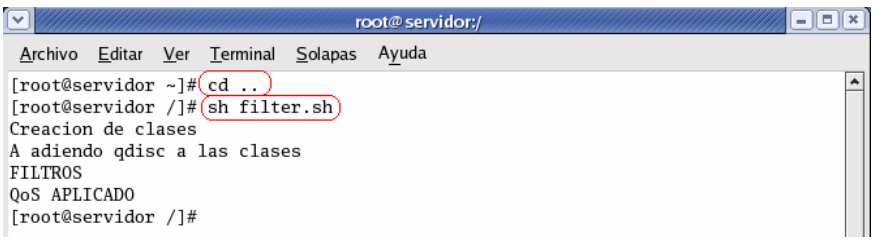


Figura 4-150: Ejecución del archivo filter.sh

4.8 Prueba de Servidores y QoS

Para realizar las pruebas de los servidores se han utilizado diferentes herramientas incluidas en los sistemas operativos tanto Linux como Windows a continuación se detalla las pruebas realizadas a cada uno se los servidores.

4.8.1 Prueba del servidor DNS

Tabla 4-4: Resultados de las pruebas a los dominios creados con nslookup

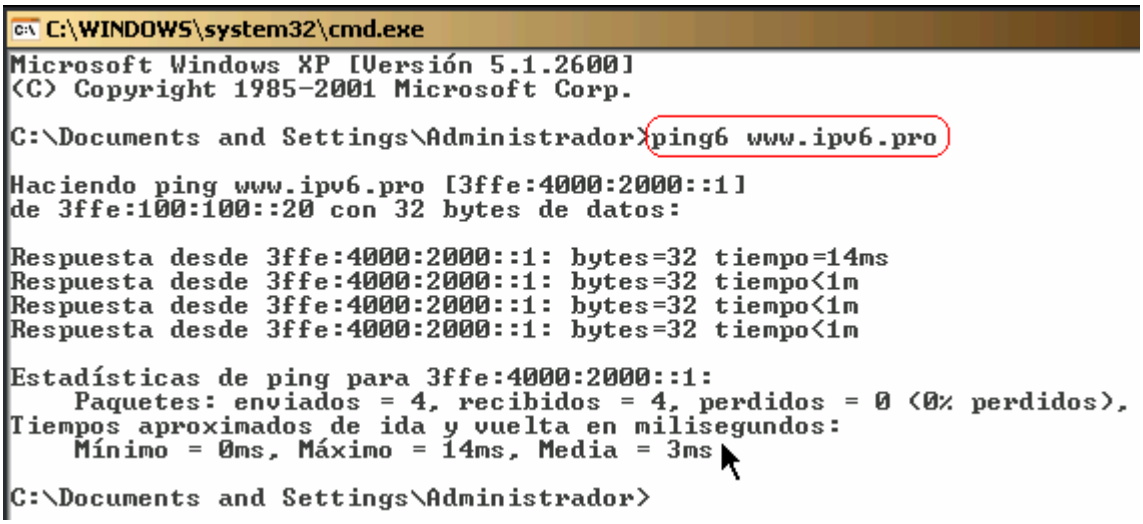
Código: PSD1	**Nombre:** nslookup
Descripción: Para comprobar que el dominio creado *www.ipv6.pro* está respondiendo se hizo pruebas utilizando el comando *nslookup* desde una maquina con Windows XP para lo cual se ha utilizado el símbolo de sistema para introducir este comando.	
Ambiente: Windows XP SP2	
Resultados:	

```
C:\Users\usuarioX>nslookup
Servidor predeterminado:  servidor.ipv6.pro
Address:  [3ffe:4000:2000::1]:53

> set type=any
> ipv6.pro
Servidor:  servidor.ipv6.pro
Address:  [3ffe:4000:2000::1]:53

ipv6.pro        MX preference = 5, mail exchanger = 3ffe:4000:2000::1.ipv6.pro
ipv6.pro
        primary name server = servidor
        responsible mail addr = javier.ipv6.pro
        serial  = 1170612678
        refresh = 10800 (3 hours)
        retry   = 3600 (1 hour)
        expire  = 604800 (7 days)
        default TTL = 38400 (10 hours 40 mins)
ipv6.pro        nameserver = servidor
> _
```

Tabla 4-5: Resultados de las pruebas a los dominios creados utilizando ping

Código: PSD2	**Nombre:** ping
Descripción: Para comprobar que el dominio creado *www.ipv6.pro* está respondiendo se hizo pruebas utilizando el comando *ping6* desde una maquina con Windows XP SP2 para lo cual se ha utilizado el símbolo de sistema para introducir este comando.	
Ambiente: Windows XP SP2	
Resultados: 	

4.8.2 Prueba del servidor de Correos

Tabla 4-6: Verificación de los puertos de escucha de correo con netstat

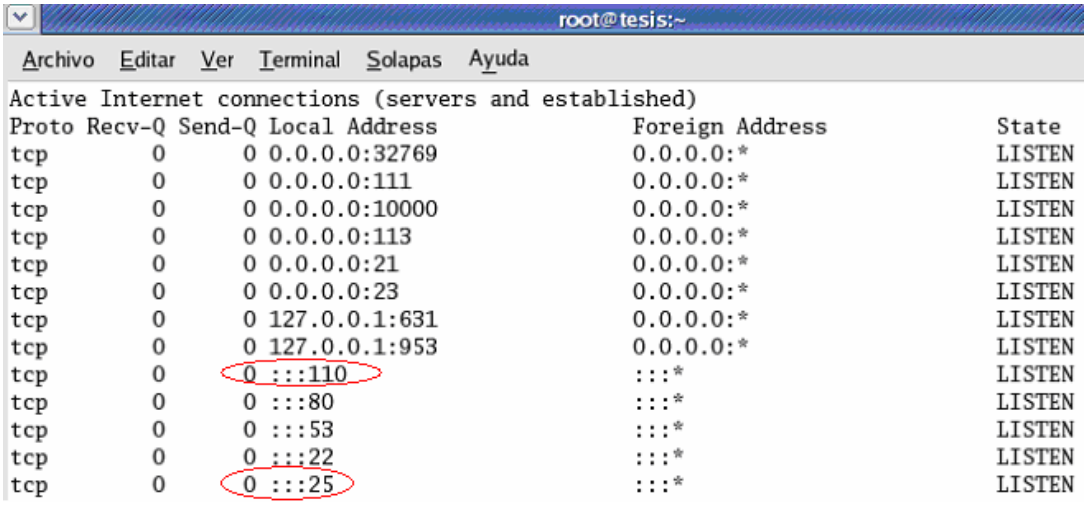
Código: PSC1	**Nombre:** netstat
Descripción: Para verificar que el servidor de correos este bien configurado se debe observar que los puertos 25 y 110 estén abiertos y escuchando en el servidor para lo cual en una ventana de Terminal introduciremos el comando *netstat –na*	
Ambiente: Centos 4.3	
Resultados: 	

Tabla 4-7: Pruebas de envió de correo mediante telnet

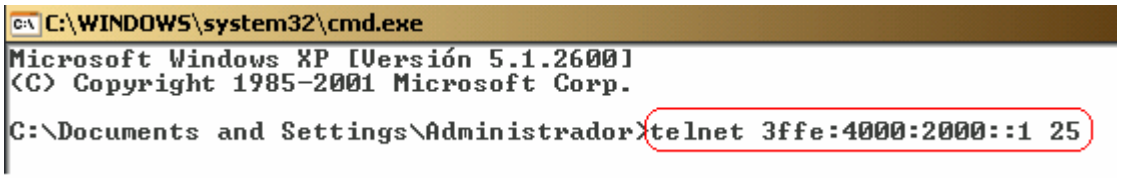
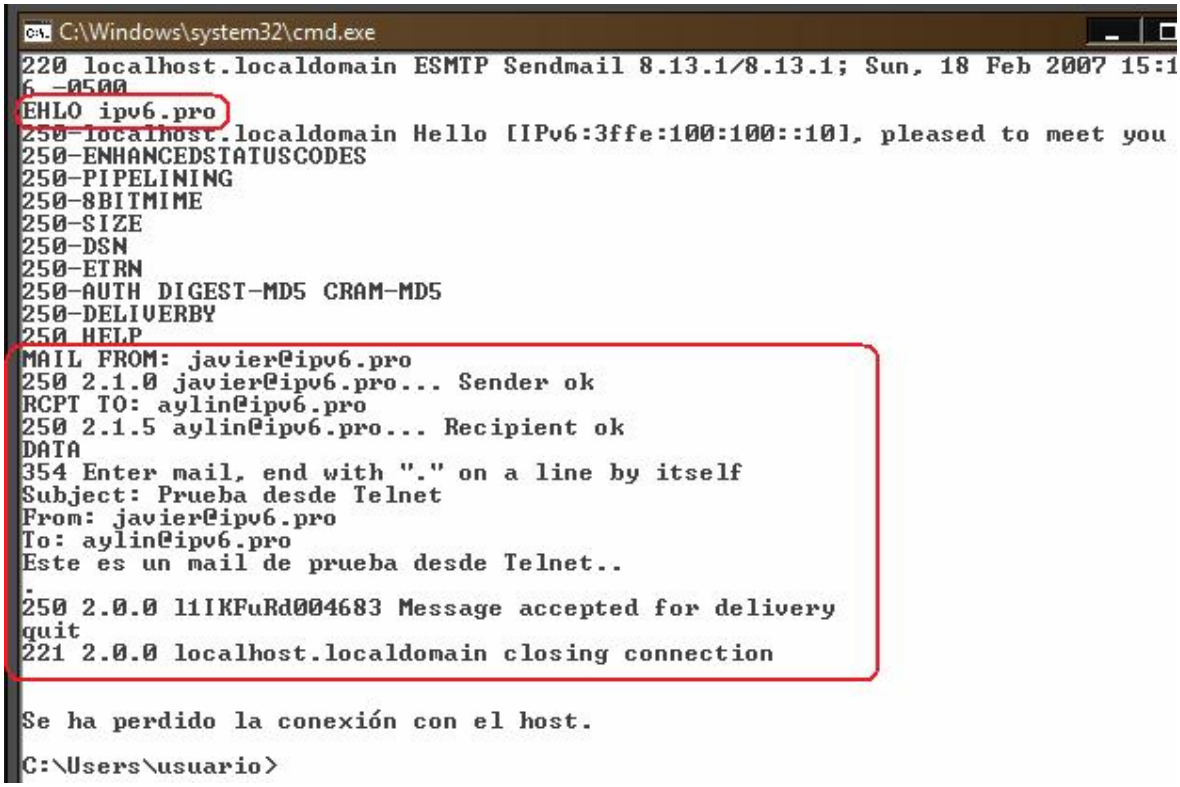
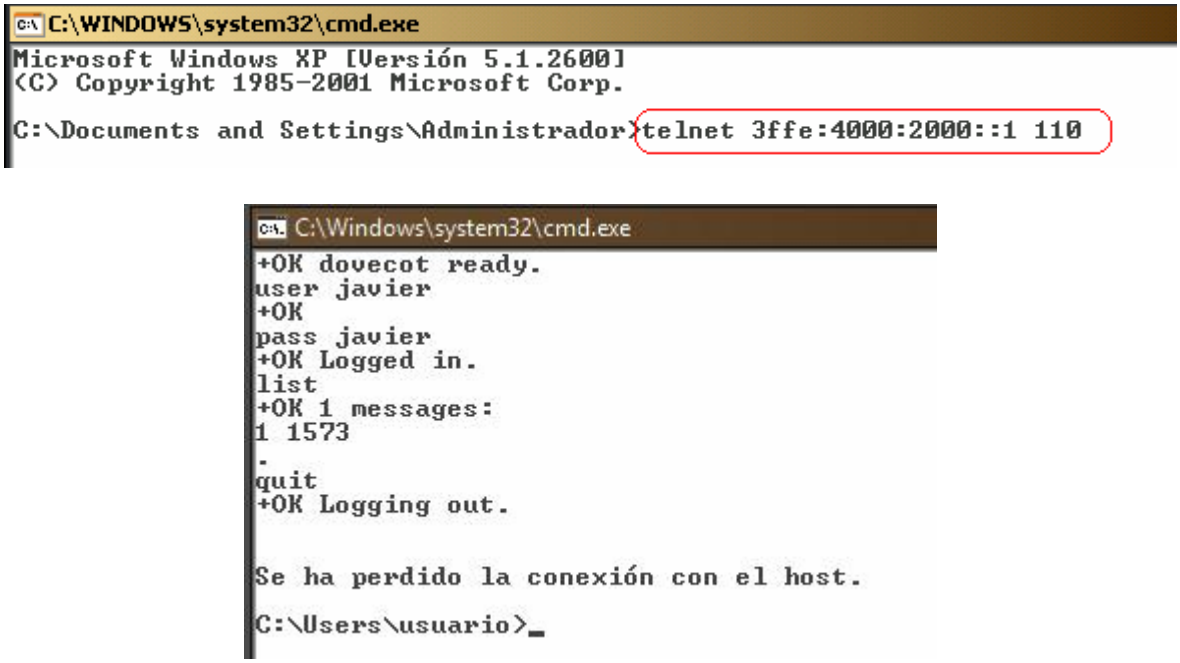
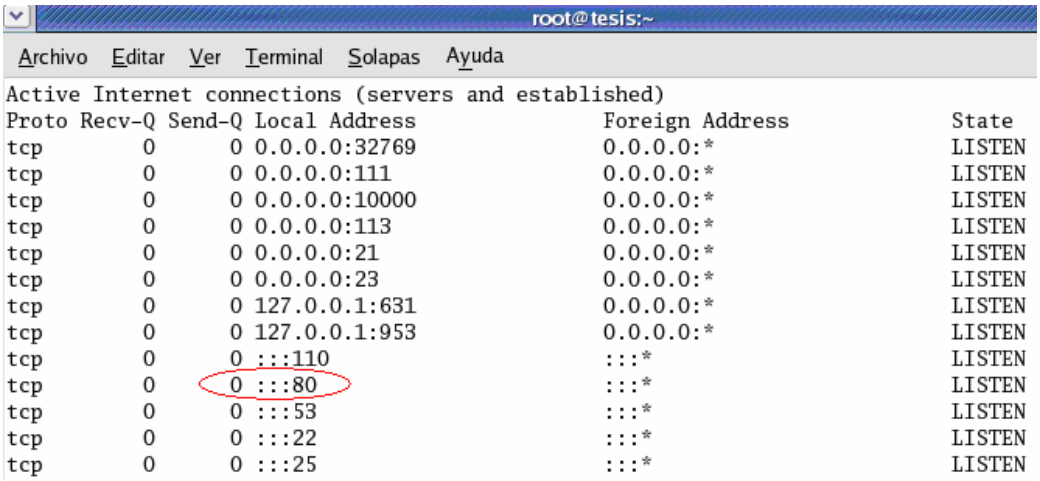
Código: PSC2	**Nombre:** telnet
Descripción: Para las pruebas se ha utilizado el cliente *telnet* de Windows XP para lo cual primero establecemos una conexión entre el cliente y el servidor tecleando en la línea de comandos **telnet** *direccion_ip puerto* ; luego de establecida la conexión introducimos el comando **EHLO** *dominio*	
Ambiente: Windows XP	
Resultados: **Envió de mail mediante telnet**   	

Tabla 4-8: Pruebas de recepción de correo mediante telnet

Código: PSC3	**Nombre:** telnet
Descripción: Para las pruebas se ha utilizado el cliente telnet de Windows XP SP2 para lo cual primero establecemos una conexión entre el cliente y el servidor tecleando en la línea de comandos **telnet** *direccion_ip puerto* ; luego de establecida la conexión introducimos el user y password de la cuenta de correo a la que queremos tener acceso.	
Ambiente: Windows XP SP2	
Resultados: **Recepción de mail mediante telnet**  	

4.8.3 Prueba del servidor Web

Tabla 4-9: Verificación del puerto de escucha de apache mediante netstat

Código: PSW1	**Nombre:** netstat
Descripción: Para verificar que el servidor Web este bien configurado se debe observar que el puerto 80 este abierto y escuchando en el servidor para lo cual en una ventana de Terminal se ha tecleado el comando netstat –na	
Ambiente: Windows XP SP2	
Resultados: 	

4.8.4 Prueba del árbol de QoS

Tabla 4-10: Verificación del árbol de QoS

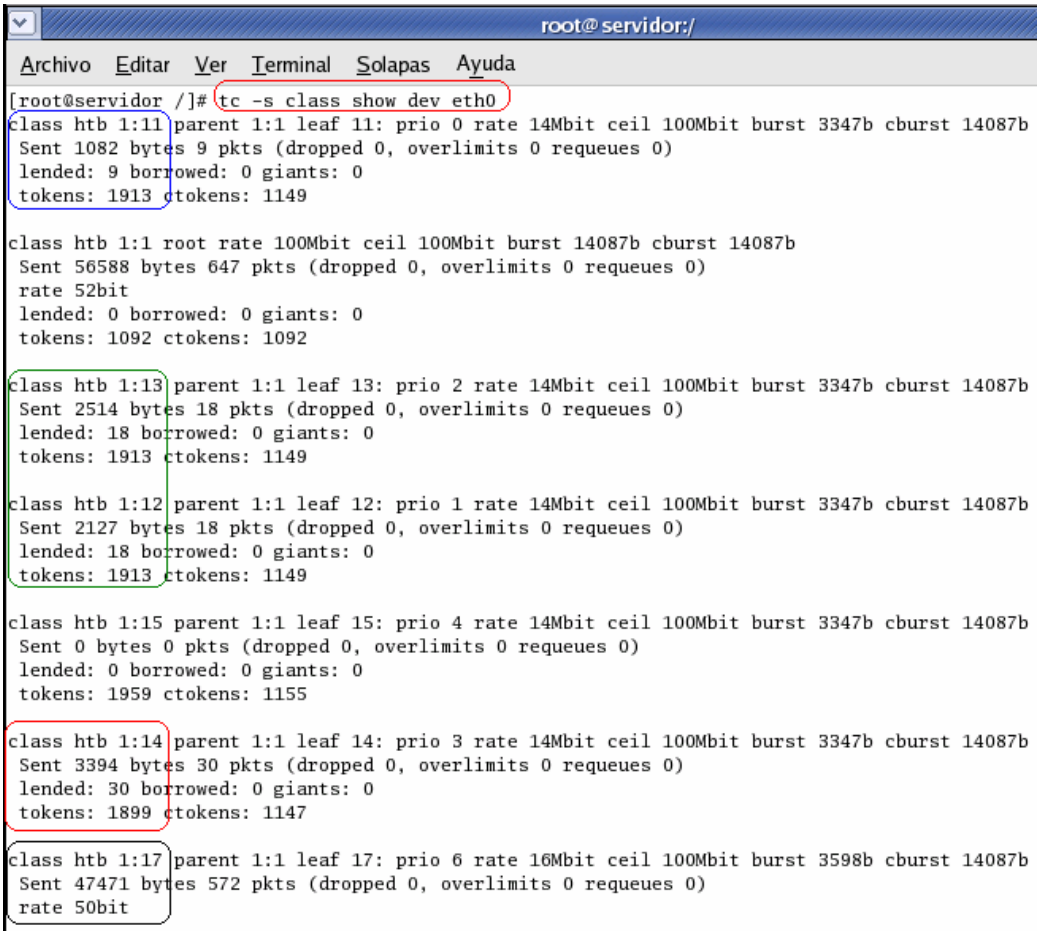
Código: PQoS	**Nombre:** Árbol de QoS
Descripción: Para verificar que cada una de las clases esta recibiendo el trafico correspondiente al puerto utilizado por la aplicación del cliente se realizo un despliegue de estadísticas de las clases en una ventana de Terminal mediante el comando ***tc –s class show dev eth0*** este despliegue de estadísticas se lo ha realizado en el servidor y desde la maquina cliente generamos trafico correspondiente a cada clase mediante las siguientes aplicaciones: Evolution → Trafico al puerto 25 y 110 → Clase 1:12 y 1:13 Mozilla Firefox → Trafico al puerto 80 y 21 → Clase 1:11 y 1:14 Resto de trafico → Clase 1:17 Por lo tanto las clases 1:15 y 1:16 no tendrán ningún tipo de trafico ya que no se ha utilizado ninguna aplicación desde el cliente que realice peticiones a los puertos 22(SSH) y 23 (Telnet) La clase 1:10 es la que recibe todo el trafico y lo distribuye según el puerto que la aplicación cliente utiliza por esto la suma de todos los paquetes enviados a las demás clases debe ser igual a los enviados por esta clase.	

Clase 1:11	1082
Clase 1:12	2514
Clase 1:13	2127
Clase 1:14	3394
Clase 1:17	47471

Clase 1:1	56588

Ambiente: Fedora Core 5

Resultados:



CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- Con IPv6 se tiene mayor velocidad en el procesamiento de paquetes en los routers dado que estos no realizan fragmentación a cada salto solo los nodos de origen son los encargados de realizar fragmentación por tanto se elimina el tiempo que tomaba este proceso con IPv4 en cada salto.

- La integración de mecanismos de seguridad, autenticación y confidenciabilidad dentro del núcleo de protocolo IPv6 es una de las grandes ventajas que este nuevo protocolo presenta.

- El encabezado de IPv6 maneja QoS mediante el campo clase de trafico por lo que se puede identificar y controlar el trafico utilizando los conceptos involucrados con Servicios diferenciados razón por la cual el manejo de trafico en IPv6 se lo hace de manera mas eficiente que en IPv4.

- El uso de un servidor DNS en redes que utilicen como protocolo el IPv6 es fundamental dado que las direcciones son mas difíciles de recordar que las de IPv4 puesto que están compuestas de 8 agrupaciones de 16 bits.

- Utilizar routers a base de PC's de escritorio con sistema operativo GNU/Linux y zebra como software de roteo resulta para pequeñas y

medianas empresas mucho mas económico y brinda mas beneficios que utilizar un router comercial.

- Mediante la utilización de QoS se puede realizar un control de tráfico adecuado dentro de una red local permitiendo priorizar el tráfico que requiera de más recursos para su correcto funcionamiento y rendimiento.

5.2 Recomendaciones

- Apoyar e impulsar las iniciativas de desarrollo de aplicaciones que utilicen IPv6 como protocolo de comunicación.

- Promover la implementación de redes y servicios que trabajen conjuntamente con IPv4 e IPv6 en instituciones educativas para investigar y aportar con nuevas ideas al nuevo protocolo IPv6.

- Realizar una investigación a fondo acerca de las ventajas que presenta IPv6 en las comunicaciones con dispositivos móviles.

- Explorar el impacto que tendrá el uso del protocolo IPv6 en las seguridades informáticas.

- Efectuar un software que automatice el control de tráfico en GNU/Linux.

BIBLIOGRAFIA

Textos

Comer,D. (2000). "Internetworking with TCP/IP Vol 1", Cuarta Edición, Prentice Hall.

Deering,S. & Hiden, R.(2000). "IPv6". Primera Edición.

Thomson,S & T. Narten. (2000). "IPV6 Stateless Address Autoconfiguration", Primera Edición.

D. Johnson,C. Perkins, J Arkko.(2004). "Mobility Support in IPv6", Primera Edición

Bandel David. (2006). "LINUX", Sexta Edición, Prentice Hall.

Paginas Web

Web Oficial de IPv6	http://www.ipv6.org
Linux advance routing & Trafic Control	http://www.lartc.net
Internet Engineering Task Force IETF	www.ietf.org
Request for Comments RFC	http://www.rfc-es.org/
Internet Architecture Board IAB	http://www.isi.edu/iab/
Ultimas versiones del kernel	http://www.kernel.org
Ultimas versiones de IPTABLES	http://www.netfilter.org

ANEXO A

Manual para compilar el kernel y comandos de Router CISCO

Manual para compilar el kernel

Los pasos a seguir son:

a) Obtener e instalar las fuentes del kernel

b) Configurar el kernel

c) Compilar

d) Instalar

e) Probar

a) Obtener e instalar las fuentes del kernel

Se debe obtener las fuentes del kernel que se quiere instalar (paquete kernel-source) estas fuentes se las puede bajar de Internet (www.kernel.org). Si se obtiene un archivo comprimido se lo debe descomprimir, al final se debe tener un directorio /usr/src/linux con los fuentes. También se puede instalarlo con otro nombre (p.e.kernel-x.x.xx) y luego crear un enlace simbólico al directorio linux. Al final debe existir /usr/src/linux y contener toda la estructura de directorios y archivos de los fuentes del kernel.

b) Configurar el kernel

Se sitúa en el directorio linux (#cd /usr/src/linux) y se teclea make menuconfig (o make xconfig para entorno X). Con esto se accede al menú de configuración del kernel. Aquí se debe configurar un montón de cosas, desde modelo y número de procesadores que tiene el sistema, hasta el sistema de archivos, hardware del equipo, etc. Muchas de las opciones pueden activarse como estáticas o como módulos. Siempre que se pueda se debe elegir la opción de módulos para no exceder en el tamaño del kernel.

c) Compilar

Las opciones comúnmente usadas son las siguientes:

#make dep (chequea las dependencias)
#make clean
#make bzImage (crea un kernel compacto. Existen otras opciones como make zImage)
#make modules (compila los módulos)
#make modules_install (instala los módulos compilados)

Al finalizar make bzImage dará un mensaje indicando que se creo el kernel y el tamaño que este tiene.

d) Instalar

Antes que nada es preferible hacer una copia del kernel actual renombrando los archivos /boot/system.map-2.4.18 y /boot/bzImage-2.4.18 como /boot/system.map-old y /boot/bzImage-old

Ahora se debe instalar el nuevo kernel, esto depende del modo en que se este arrancando Linux. Si se arranca desde un disquete se hará lo siguiente:

- Se copia y renombra el archivo /usr/src/linux/system.map en /boot/system.map 2.4.18
- Se copia y renombra el archivo /usr/src/linux/arch/i386/boot/bzImage en /boot/bzImage-2.4.18

En este punto se tiene en /boot los archivos system.map-2.4.18 y bzImage-2.4.18 que son el nuevo kernel compilado. Ahora se crea un nuevo disquete de arranque mediante el comando #mkdisk/boot/bzImage-2.4.18. Este comando crea el archivo imagen del kernel vmlinux en el disquete.

e) Probar

Ahora se reinicia el equipo y se prueba el nuevo kernel. Si hay problemas, se arranca desde el disquete anterior y se reconfigura.

Comandos de Routers CISCO

show interfaces : muestra información sobre las interfaces del enrutador

show interface [interfaz] : muestra información de una interfaz específica

show versión : muestra información de la configuración de hardware

show protocols : lista los protocolos de red configurados actuales

show processes : muestra información sobre la utilización de la CPU

show cdp neighbor : muestra los vecinos de interconexión

show running-config : muestra la configuración actual que se ejecuta

show startup-config : muestra la configuración de arranque del enrutador

show logging : muestra los logs generados y almacenados en memoria

show flash : muestra la cantidad de memoria flash disponible y no disponible

config terminal : establece al enrutador en modo de configuración

line console 0 : establece la conexión por consola en modo de configuración

line vty 0 4 : establece la conexión de sesiones telnet en modo de configuración

copy running-config startup-config : guarda cambios hechos a la configuración actual

setup : inicializa asistente de configuración

? : muestra las posibles opciones de comandos actuales con su explicación

^z : salir de modo de configuración

enable : instrucción para entrar en modo privilegiado

disable : sale del modo privilegiado

show ip interface brief : muestra las interfaces ip y su estado

logging synchronous : establece sincronización entre los mensajes mostrados por consola y las entradas del usuario

ANEXO B

Manual de Iptables

Una vez tengamos el kernel compilado con las opciones necesarias, ya podemos pasar a configurar nuestro firewall con iptables. Cuando definamos una regla de filtrado, deberemos especificar unos criterios que se aplicarán al paquete, y en caso de que se cumplan, un objetivo. Es decir, si llega un paquete ICMP (criterio) de la dirección IP x.y.z.w (otro criterio), elimínalo (objetivo). Esa regla cuando la definamos, la meteremos dentro de una cadena predefinida por iptables, o una definida por el propio usuario. Por ejemplo, si queremos evitar que los usuarios de la LAN hagan pings al exterior, definiríamos una regla de filtrado en la cadena de salida de esta forma: si llegan paquetes ICMP (criterio) desde cualquier IP de la LAN (criterio) con destino al exterior (cadena FORWARD), elimínalo (objetivo).
Los objetivos básicos son:

• **ACCEPT**. Acepta el paquete.

• **DROP**. Rechaza el paquete.

• **RETURN**. Cuando un paquete entra en una regla que tiene como objetivo RETURN, pueden suceder dos cosas: si la cadena es una subcadena de otra, entonces deja de examinarse la subcadena y se sigue con la cadena principal; y si la cadena es la principal y tiene como objetivo RETURN, entonces se aplicará la política por defecto, normalmente ACCEPT o DROP.

• **QUEUE**. Pasa el paquete a espacio de usuario, donde otro programa los puede recoger y analizar.

a) Tipos de tablas de Iptables

En iptables existen tres tipos de tablas, cada una con unas cadenas internas predefinidas. Dependiendo de lo que queramos hacer, necesitaremos utilizar unas tablas junto con sus cadenas. Los 3 tipos de tablas son: *filter*, *nat* y *mangle*.
En cada tabla, nos podremos encontrar con alguna de estas cadenas predefinidas:

• **INPUT**. Esta cadena se utiliza para los paquetes de entrada.
• **OUTPUT**. Esta cadena se utiliza para los paquetes salientes.
• **FORWARD**. Si los paquetes van destinados a otra máquina que no sea la local, se utilizara esta cadena.
• **PREROUTING**. Se utiliza para el manejo de paquetes antes de que sean enrutados.
• **POSTROUTING**. El manejo de paquetes se realiza después de su enrutamiento.

El *forwarding* debe estar activado para que podamos utilizar la tabla FORWARD, de lo contrario, todo paquete que tenga un destino distinto de la máquina local será eliminado. Activaremos el forwarding mediante:

```
zen:~# echo 1 > /proc/sys/net/ipv4/ip_forward
zen:~# echo  > /proc/sys/net/ipv6/conf/all/forwarding
```

a.a) Tabla filter

Esta es la tabla por defecto de iptables. Es la que utilizaremos para definir reglas de filtrado. Sus cadenas son:

• INPUT

• OUTPUT

• FORWARD

y sus objetivos son:

• ACCEPT

• DROP

• QUEUE

• RETURN

a.b) Tabla nat

Cuando queramos utilizar cualquier tipo de NAT, esta será la tabla a utilizar. Sus cadenas son:

• PREROUTING

• OUTPUT

• POSTROUTING

y sus objetivos son:

• DNAT

• SNAT

• MASQUERADING

a.c) Tabla mangle

En esta tabla podremos modificar parámetros de los paquetes como el TOS (Type Of Service) de IPv4. Está será la tabla que utilizaremos para el marcado de paquetes y posterior priorización para aplicar QoS. Esta tabla contiene todas las cadenas anteriores:

• INPUT

• OUTPUT

• FORWARD

• PREROUTING

• POSTROUTING

Sus principales objetivos son:

• TOS

• MARK

b) Parámetros

Con los siguientes parámetros podremos definir reglas en base a determinadas características.

• **-p**: Tipo de protocolo a tratar entre: *tcp*, *udp*, *icmp* o *all* (los tres).

• **-s**: Dirección origen.

• **-d**: Dirección destino.

• **-i**: Interfaz de entrada.

• **-o**: Interfaz de salida.

• **-j**: Objetivo para una regla.

Nota: Tanto en los parámetros como en las extensiones de coincidencia, podemos utilizar el símbolo **!** para invertir la regla. Este ejemplo, aceptará cualquier protocolo que NO sea *icmp*.

```
zen:~# iptables -A INPUT -p ! icmp -j ACCEPT
```

c) Extensiones de coincidencias

Una forma de afinar nuestra configuración con iptables es usando los módulos de extensiones de coincidencias. Con la opción -p podemos especificar el protocolo a tratar como TCP, ICMP o UDP y hacer una clasificación más exhaustiva en cuanto a flags TCP, tipo de ICMP, puertos UDP, etc

Otra forma es mediante la opción -m que cargará el módulo en cuestión junto con sus opciones listas para utilizar. Esta opción es bastante útil ya que podemos utilizar tanto los módulos que vienen *de serie* con iptables, como otros que van apareciendo a posteriori.

d) Extensiones de objetivo

Esta es la parte más importante de iptables, ya que cuando tengamos un paquete que coincida con nuestras reglas, le deberemos decir que tiene que hacer con él. Sobra decir que no es lo mismo un -j ACCEPT que un –j DROP. Nos centraremos en los objetivos más comunes.

LOG. Lo utilizaremos cuando queramos que quede constancia en el syslogd (puede ser otro) del paso de algún paquete. Básicamente se puede utilizar para el *debuging* del firewall, o simplemente si queremos tener constancia de un conato de ataque.

```
snowball:~# iptables -A INPUT -p icmp -j LOG --log-prefix "PAQUETES ICMP FILTRADOS "
--log-level debug --log-ip-option
```

Aquí definimos una regla para los paquetes ICMP (-p icmp) de entrada (-A INPUT). Cuando el sistema reciba paquetes ICMP, entonces iptables logeará (-j LOG) a los paquetes. Con --log-prefix conseguiremos que cuando logee los paquetes añadan el prefijo indicado en los logs. Esto es especialmente útil si tenemos un archivo grande de logs, en donde luego con grep podremos clasificar fácilmente los logs que nos interesen. Con --log-level le indicamos el rigor con el que queremos que guarde el log (man syslog.conf para más opciones). Por último, le indicamos que queremos que guarde todas las opciones de la cabecera IP con --log-ip-option.

Podemos comprobar que la regla funciona haciendo pings a la máquina en cuestión.

```
snowball:~# ping -c 1 localhost
snowball:~# tail /var/log/syslog

Mar 21 11:58:38 snowball kernel: PAQUETES ICMP FILTRADOS IN=lo
OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1
```

DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=IPv6/IPv4 ID=0 DF

PROTO=ICMP TYPE=8 CODE=0 ID=2082 SEQ=1

Mar 21 11:58:38 snowball kernel: PAQUETES ICMP FILTRADOS IN=lo

OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.0.1

DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=IPv6/IPv4 ID=28729

PROTO=ICMP TYPE=0 CODE=0 ID=2082 SEQ=1

En caso de querer que los logs aparezcan en /var/log/syslog , deberíamos modificar /etc/syslog.conf.

MARK. Lo utilizaremos para el marcaje de paquetes. Este será el objetivo que se utilizará junto con tc para definir las reglas de QoS. Cuando marcamos un paquete no estamos cambiando ningún valor de la cabecera IP, sino el valor del paquete de cara al kernel. El valor que tomará MARK será un entero entre 0 y 65535.

snowball:~# iptables -t mangle -A PREROUTING -p tcp --dport 143 -j MARK --set-mark 2

Le indicamos a la tabla mangle que todo paquete IMAP (-p tcp --dport 143) que reciba la máquina deberá ser marcado (-j MARK --set-mark 2) para su posterior clasificación y envió (-A PREROUTING).

TOS. Lo utilizaremos para cambiar el valor de 8 bits del campo TOS dentro de la cabecera IPv4. Este campo indica la prioridad del tráfico, y se aplicará en routers que tengan soporte para QoS. Los valores son:

• Minimize-Delay - 0x10

• Maximize-Throughput - 0x08

• Maximize-Reliability - 0x04

• Minimize-Cost 2 - 0x02

• Normal-Service 0 - 0x00

Un ejemplo sería:

```
snowball:~# iptables -t mangle -A PREROUTING -p icmp -j TOS --set-tos Minimize-Delay
```

Marcamos todos los paquetes ICMP con el TOS Minimize-Delay. También los podríamos haber marcado con 0x10. Podemos comprobar que ha cambiado el valor del TOS haciendo un ping y capturando el paquete con un sniffer.

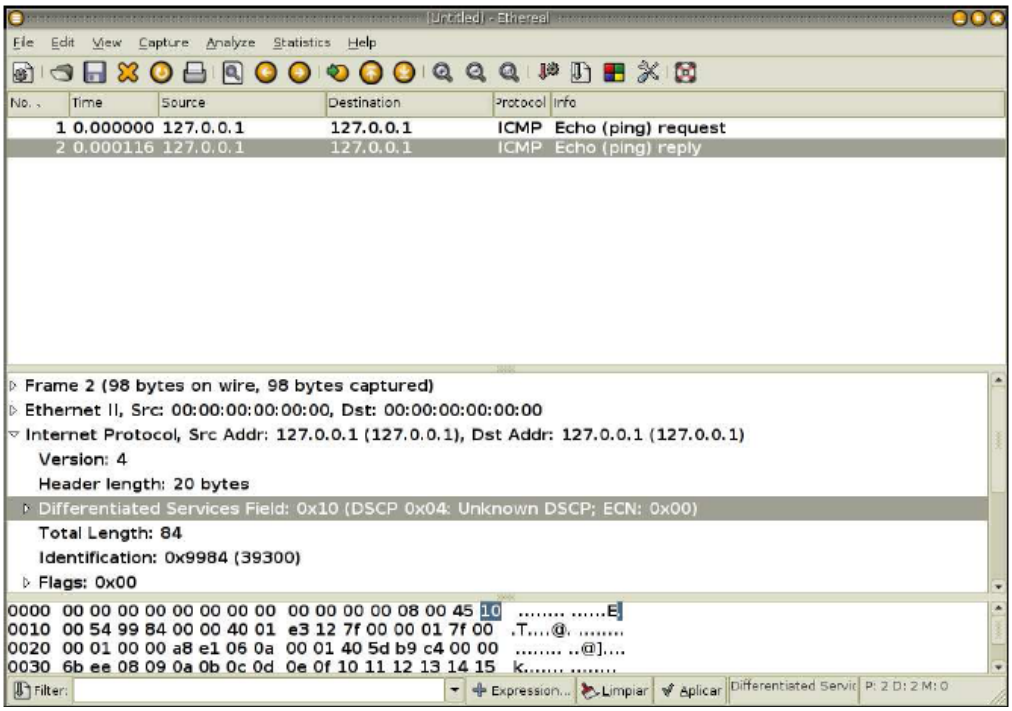


Figura b-1: Captura de paquete con Ethereal

REJECT. Es igual que DROP, pero más "educado". Cuando descarta el paquete puede devolver un error ICMP como *icmp-net-unreachable, icmp-host-unreachable, icmp-port-unreachable, icmp-proto-unreachable, icmp-net-prohibited o icmp-host-prohibited*.

snowball:~# iptables -A INPUT -p icmp -j REJECT --reject-with icmp-host-unreachable

snowball:~# ping -c 1 localhost

PING localhost (127.0.0.1) 56(84) bytes of data.

--- localhost ping statistics ---

1 packets transmitted, 0 received, 100% packet loss, time 0ms

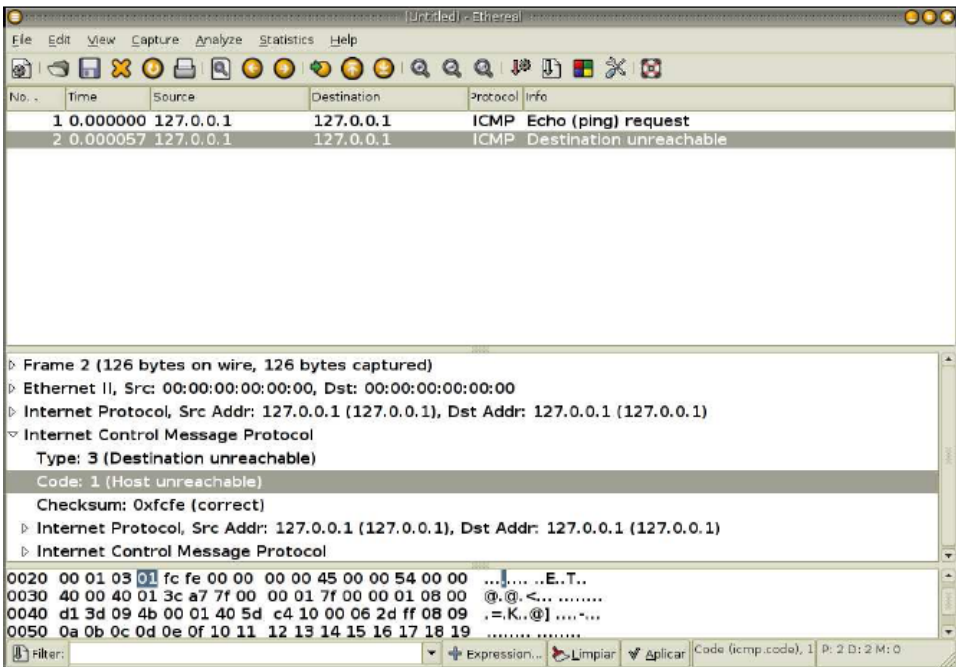


Figura b-2: Mensaje de error ICMP

DNAT. Objetivo válido para la tabla nat y las cadenas PREROUTING y OUTPUT. Con DNAT, podemos cambiar la dirección de destino del paquete.

```
snowball:~# iptables -t nat -A PREROUTING -p tcp -d 81.65.123.90 --dport 80  \
-j DNAT --to-destination 192.168.1.10:80
```

Cuando recibimos (-A PREROUTING) una petición web (-p tcp --dport 80) a la dirección IP pública (-d 81.65.123.90), la enviamos a la dirección IP privada de la LAN (--to-destination 192.168.1.10:80).

REDIRECT. Este objetivo sólo es válido en las cadenas PREROUTING y OUTPUT de la tabla nat. Es una especialización de DNAT. En DNAT podemos elegir la IP de destino, en REDIRECT siempre será la del propio host.

```
snowball:~# iptables iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT  \
--to-ports 8080
```

Cuando recibimos (-A PREROUTING) una petición web (-p tcp --dport 80), la redirigimos al puerto del *proxy* (-j REDIRECT --to-ports 8080).

SNAT. Objetivo valido también en la tabla nat, pero únicamente en la cadena POSTROUTING. Este objetivo es muy utilizado sobretodo para todos aquellos que tiene una única conexión a Internet y varias máquinas detrás.

```
snowball:~# iptables iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source 81.65.123.90
```

Cuando un paquete de la LAN es enrutado (-A POSTROUTING) hacia la ethernet que tiene salida a Internet (-o eth0), entonces cambia la IP original, por la que nos

suministra nuestro ISP (-j SNAT --to-source 81.65.123.90) para poder tener salida a Internet.

MASQUERADE. Es una particularización de SNAT, y por tanto únicamente utilizable en la cadena POSTROUTING de la tabla nat. Se utiliza en caso de que la dirección IP pública sea asignada dinámicamente (que suele ser lo más normal), con lo que nos ahorramos la opción --to-source.

```
snowball:~# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Este es el ejemplo más utilizado a nivel mundial .Realizará la misma tarea que el SNAT, pero no hará falta que le indiquemos IP. En caso de tener una conexión a Internet con una dirección IP fija asignada por nuestro ISP, utilizaremos SNAT.

e) Manejo de las reglas

Las operaciones básicas que se pueden hacer a una cadena son:

- **-N** Crea una nueva cadena
- **-X** Borra una cadena definida por el usuario
- **-A** Añade una regla nueva al final de la cadena
- **-D** Elimina reglas de una cadena
- **-I** Inserta una regla en la posición de la cadena que le indiquemos
- **-R** Mueve la regla dentro de la cadena
- **-F** Elimina TODAS las reglas de la cadena
- **-L** Lista todas las reglas de la cadena
- **-P** Cambia la política de cada cadena

ANEXO C

CAPTURA DE PAQUETES CON ETHEREAL

En este anexo se presenta una serie de pruebas realizadas a los servicios levantados estas pruebas consisten en capturar los diferentes paquetes que circulan por la red mediante la herramienta Ethereal.La captura de los paquetes se la ha hacho en el servidor y las peticiones se las ha hecho utilizando diferentes programas desde el cliente PC1.

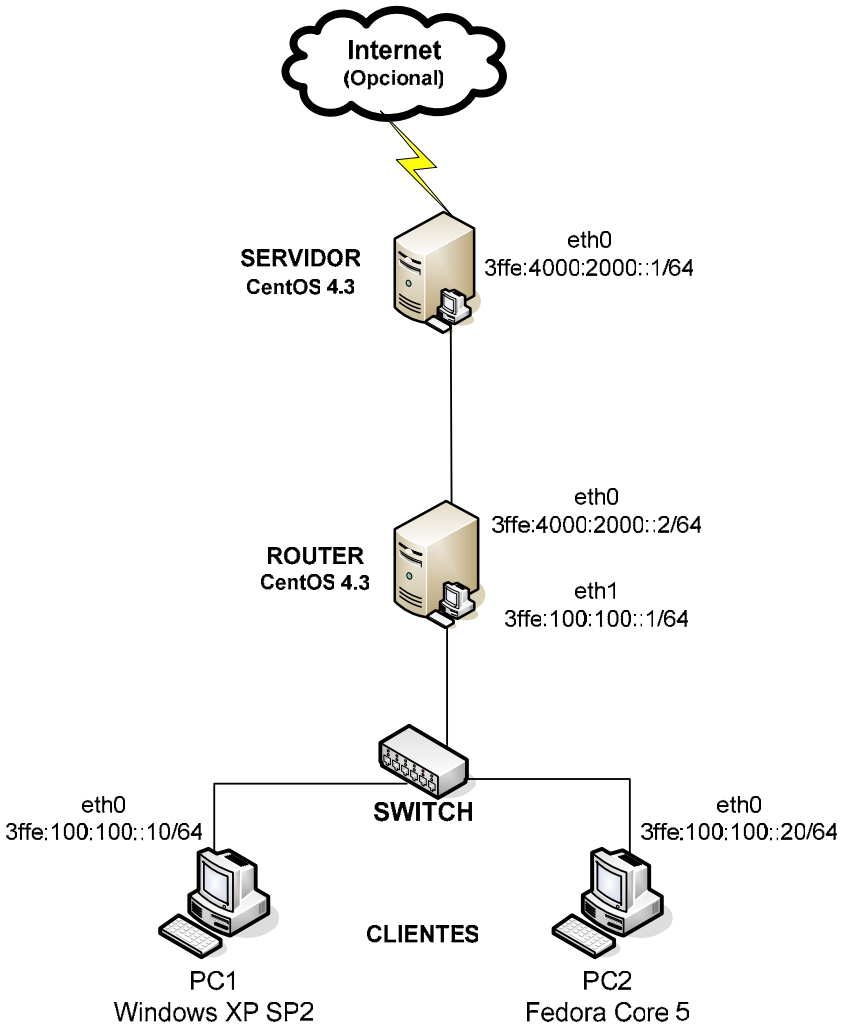


Figura c-1: Esquema de la intranet de pruebas

Tabla c-1: Captura de paquetes al realizar una petición al servidor DNS

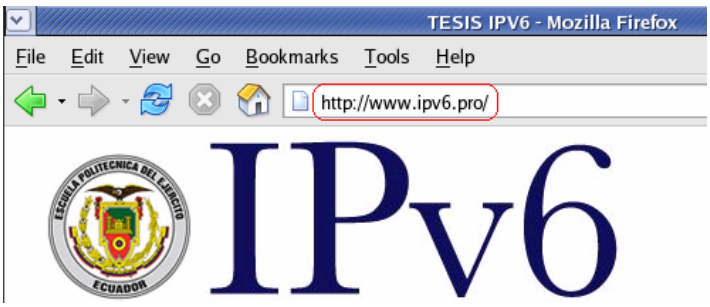
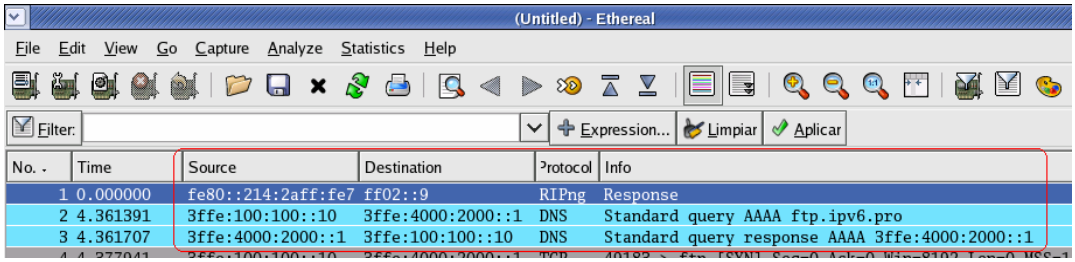
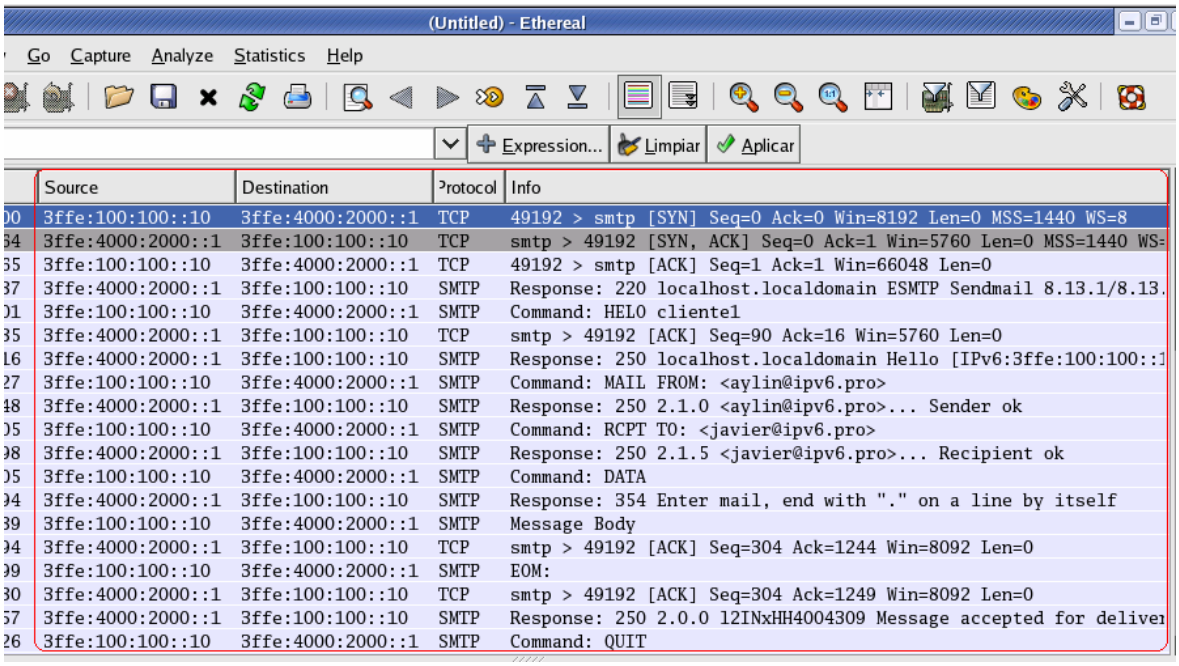
Código: PSDNS	**Nombre:** DNS
Descripción: Para esta prueba se abrio una pagina Web utilizando Mozilla Firefox desde el PC1 cuya direccion IP es 3ffe:100:100::10 la cual realiza una petición al servidor DNS cuya direccion es 3ffe:4000:2000::1 para resolver la direccion asociada a *www.ipv6.pro*	
Ambiente: Centos 4.3	
Resultados: **CLIENTE**  **SERVIDOR** 	

Tabla c-2: Captura de paquetes al realizar una petición al servidor SMTP

Código: PSSMTP	**Nombre:** SMTP
Descripción: Desde el PC1 cuya direccion IP es 3ffe:100:100::10 al realizar un envio de correo con el programa Evolution desde la cuenta *aylin@ipv6.pro* se puede observar que primero se realiza un proceso de sincronizacion entre el cliente y el servidor para después utilizar el protocolo SMTP para enviar el mail desde la cuenta aylin@ipv6.pro hacia javier@ipv6.pro	
Ambiente: Centos 4.3	
Resultados:	

ENVIO DE CORREO



Source	Destination	Protocol	Info
3ffe:100:100::10	3ffe:4000:2000::1	TCP	49192 > smtp [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1440 WS=8
3ffe:4000:2000::1	3ffe:100:100::10	TCP	smtp > 49192 [SYN, ACK] Seq=0 Ack=1 Win=5760 Len=0 MSS=1440 WS=
3ffe:100:100::10	3ffe:4000:2000::1	TCP	49192 > smtp [ACK] Seq=1 Ack=1 Win=66048 Len=0
3ffe:4000:2000::1	3ffe:100:100::10	SMTP	Response: 220 localhost.localdomain ESMTP Sendmail 8.13.1/8.13.
3ffe:100:100::10	3ffe:4000:2000::1	SMTP	Command: HELO cliente1
3ffe:4000:2000::1	3ffe:100:100::10	TCP	smtp > 49192 [ACK] Seq=90 Ack=16 Win=5760 Len=0
3ffe:4000:2000::1	3ffe:100:100::10	SMTP	Response: 250 localhost.localdomain Hello [IPv6:3ffe:100:100::1
3ffe:100:100::10	3ffe:4000:2000::1	SMTP	Command: MAIL FROM: <aylin@ipv6.pro>
3ffe:4000:2000::1	3ffe:100:100::10	SMTP	Response: 250 2.1.0 <aylin@ipv6.pro>... Sender ok
3ffe:100:100::10	3ffe:4000:2000::1	SMTP	Command: RCPT TO: <javier@ipv6.pro>
3ffe:4000:2000::1	3ffe:100:100::10	SMTP	Response: 250 2.1.5 <javier@ipv6.pro>... Recipient ok
3ffe:100:100::10	3ffe:4000:2000::1	SMTP	Command: DATA
3ffe:4000:2000::1	3ffe:100:100::10	SMTP	Response: 354 Enter mail, end with "." on a line by itself
3ffe:100:100::10	3ffe:4000:2000::1	SMTP	Message Body
3ffe:4000:2000::1	3ffe:100:100::10	TCP	smtp > 49192 [ACK] Seq=304 Ack=1244 Win=8092 Len=0
3ffe:100:100::10	3ffe:4000:2000::1	SMTP	EOM:
3ffe:4000:2000::1	3ffe:100:100::10	TCP	smtp > 49192 [ACK] Seq=304 Ack=1249 Win=8092 Len=0
3ffe:4000:2000::1	3ffe:100:100::10	SMTP	Response: 250 2.0.0 l2INxHH4004309 Message accepted for delive
3ffe:100:100::10	3ffe:4000:2000::1	SMTP	Command: QUIT

Tabla c-3: Captura de paquetes al realizar una petición al servidor POP3

Código: PSPOP3	**Nombre:** POP3

Descripción:

Desde el PC1 cuya direccion IP es 3ffe:100:100::10 al realizar la recepcion de correo mediante el manejador de correo Evolution se puede observar que primero se realiza un proceso de sincronizacion y autentificacion entre el cliente y el servidor para después utilizar el protocolo POP para recibir los mails de la cuenta aylin@ipv6.pro

Ambiente: Centos 4.3

Resultados:

RECEPCION DE CORREO

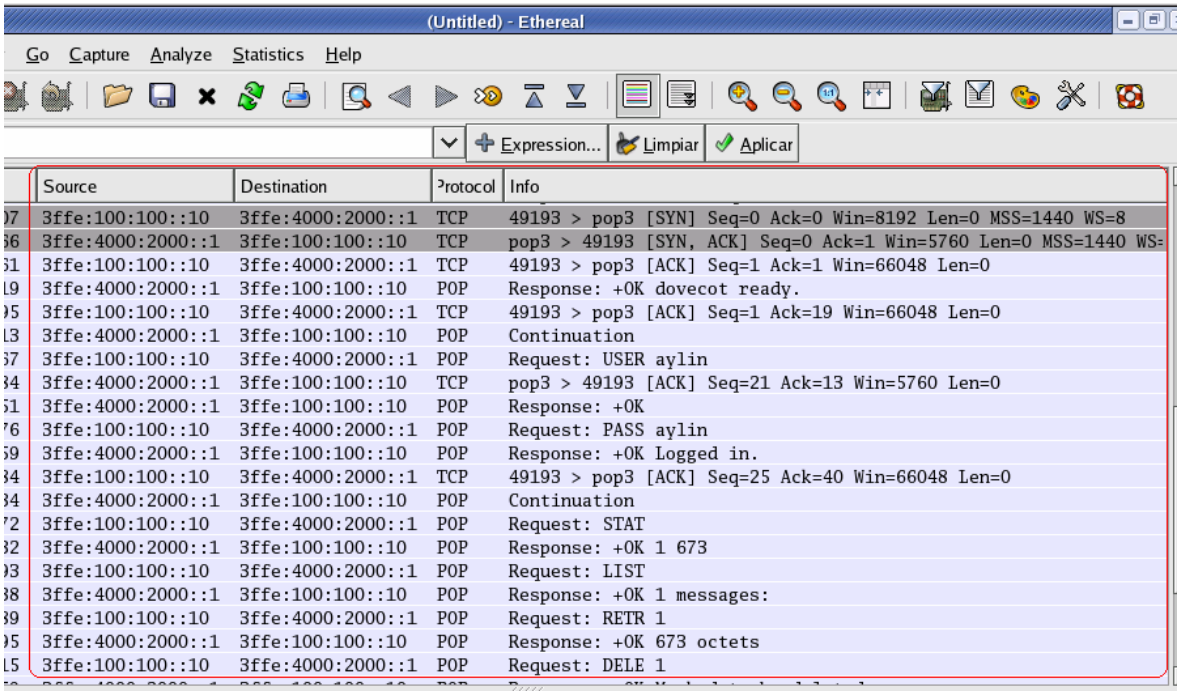
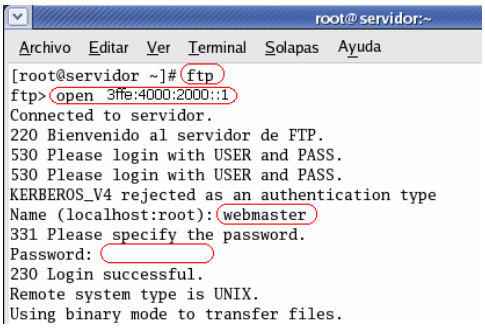
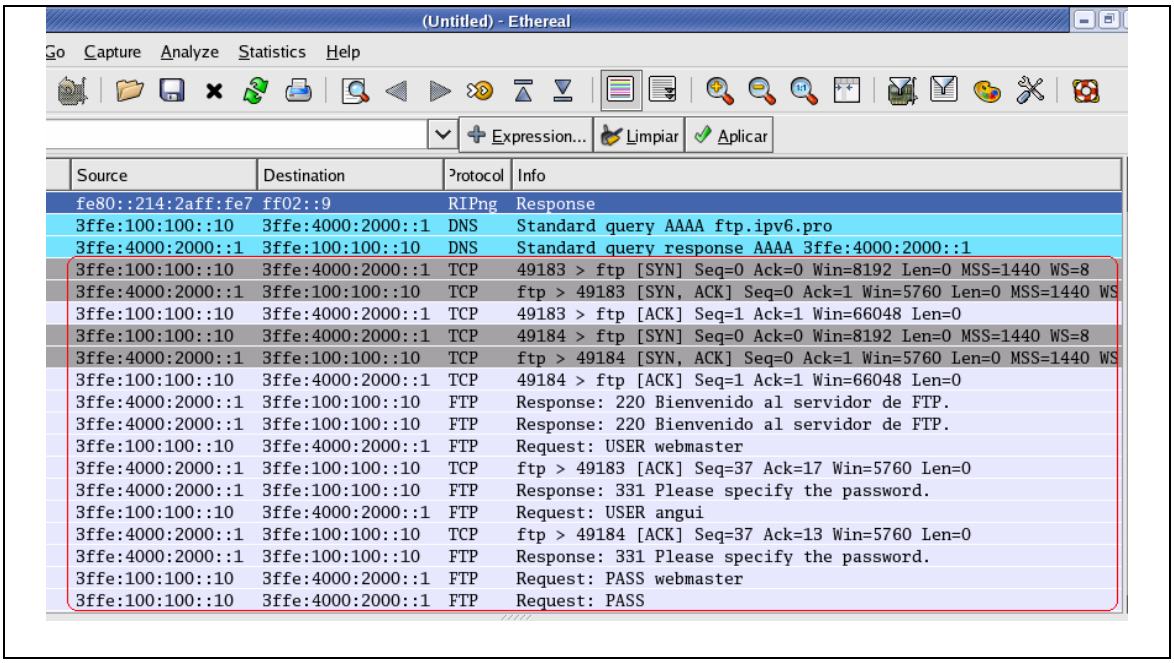


Tabla c-4: Captura de paquetes al realizar una petición al servidor FTP

Código: PSFTP	**Nombre:** FTP
Descripción: Desde el PC1 utilizando una ventana de Terminal tecleamos *ftp* para conectarnos mediante FTP hacia el servidor, para la autentificación se ha utilizado el usuario *webmaster* cuyo password es tambien *webmaster*. En la captura de paquetes hecha en el servidor vemos que que primero se realiza un proceso de sincronizacion y luego la autentificación del usuario (webmaster) para lo cual se utiliza el protocolo FTP para establecer la conexión entre el servidor y el cliente.	
Ambiente: Centos 4.3	
Resultados: **CLIENTE**  **SERVIDOR**	


(Untitled) - Ethereal
Go Capture Analyze Statistics Help
Expression... Limpiar Aplicar
Source Destination Protocol Info
fe80::214:2aff:fe7 ff02::9 RIPng Response
3ffe:100:100::10 3ffe:4000:2000::1 DNS Standard query AAAA ftp.ipv6.pro
3ffe:4000:2000::1 3ffe:100:100::10 DNS Standard query response AAAA 3ffe:4000:2000::1
3ffe:100:100::10 3ffe:4000:2000::1 TCP 49183 > ftp [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1440 WS=8
3ffe:4000:2000::1 3ffe:100:100::10 TCP ftp > 49183 [SYN, ACK] Seq=0 Ack=1 Win=5760 Len=0 MSS=1440 WS
3ffe:100:100::10 3ffe:4000:2000::1 TCP 49183 > ftp [ACK] Seq=1 Ack=1 Win=66048 Len=0
3ffe:100:100::10 3ffe:4000:2000::1 TCP 49184 > ftp [SYN] Seq=0 Ack=0 Win=8192 Len=0 MSS=1440 WS=8
3ffe:4000:2000::1 3ffe:100:100::10 TCP ftp > 49184 [SYN, ACK] Seq=0 Ack=1 Win=5760 Len=0 MSS=1440 WS
3ffe:100:100::10 3ffe:4000:2000::1 TCP 49184 > ftp [ACK] Seq=1 Ack=1 Win=66048 Len=0
3ffe:4000:2000::1 3ffe:100:100::10 FTP Response: 220 Bienvenido al servidor de FTP.
3ffe:4000:2000::1 3ffe:100:100::10 FTP Response: 220 Bienvenido al servidor de FTP.
3ffe:100:100::10 3ffe:4000:2000::1 FTP Request: USER webmaster
3ffe:4000:2000::1 3ffe:100:100::10 TCP ftp > 49183 [ACK] Seq=37 Ack=17 Win=5760 Len=0
3ffe:4000:2000::1 3ffe:100:100::10 FTP Response: 331 Please specify the password.
3ffe:100:100::10 3ffe:4000:2000::1 FTP Request: USER angui
3ffe:4000:2000::1 3ffe:100:100::10 TCP ftp > 49184 [ACK] Seq=37 Ack=13 Win=5760 Len=0
3ffe:4000:2000::1 3ffe:100:100::10 FTP Response: 331 Please specify the password.
3ffe:100:100::10 3ffe:4000:2000::1 FTP Request: PASS webmaster
3ffe:100:100::10 3ffe:4000:2000::1 FTP Request: PASS

BIOGRAFIA

Información personal

Apellidos: Ubidia Carrión
Nombres: Aníbal Javier
Nacionalidad: Ecuatoriana
Fecha de nacimiento: 30 de Agosto de 1979
Domicilio: Miguel Pontón S13-88 y calle Oe5P
Teléfono domicilio: 2615-590
Email: jubidia@hotmail.com

Educación

Estudios Secundarios

INSTITUTO SUPERIOR CENTRAL TÉCNICO
Titulo: Bachiller Técnico Industrial en Electrónica

Diplomas y Certificaciones

Suficiencia de Ingles aprobado en la ESPE de idiomas.

Experiencia laboral

- Auxiliar de sistemas en el Hospital Dr. Enrique Garcés
- Help Desk en el Centro de Salud El Carmen.
- Instructor de lenguajes de programación y redes en el SECAP.
- Servicios como programador para SUDAMERICANA de Computación.
- Técnico del departamento de garantías en TECNOMEGA.
- Asesor técnico de Internet en SATNET.
- Servicios particulares de soporte técnico.

HOJA DE LEGALIZACION DE FIRMAS

ELABORADO POR

_____
Javier Ubidia

COORDINADOR DE LA CARRERA

_____
Ing. Ramiro Delgado

Lugar y fecha: _____